Edición digital como metodología para una edición global

Análisis, reflexiones y propuestas para el entorno editorial actual

Ramiro
Santa Ana
Anguiano





Edición digital como metodología para una edición global

Análisis, reflexiones y propuestas para el entorno editorial actual

Ramiro
Santa Ana
Anguiano



Edición digital como metodología para una edición global

Análisis, reflexiones y propuestas para el entorno editorial actual

Ramiro Santa Ana Anguiano

Edición digital como metodología para una edición global

Última actualización
2 de octubre del 2018

AUTORÍA
Ramiro Santa Ana Anguiano

REVISIÓN DE CONTENIDOS Y DISEÑO DE PORTADA
Mariana Eguaras

EDICIÓN DE «DERECHO DE AUTOR CERO»
Mariel Quirino Andrade

ENTIDADES PARTICIPANTES





Los textos que componen este libro fueron publicados por primera vez en el *blog* de Mariana Eguaras.

I. Metodologías para la edición

Edición digital como edición «desde cero»

Por iniciativa de Mariana Eguaras estaremos publicando una serie de entradas dedicadas a la **publicación digital**, sus **características, ventajas y retos**, así como **su relación con los medios impresos** y cómo estas cuestiones afectan directamente a los procesos necesarios para producir cualquier publicación.

Si bien ya tenemos planeadas el contenido de las primeras entradas, cualquier sugerencia sobre temas a tratar será bienvenida. En la medida de lo posible, la redacción no será técnica, sino afín al público en general o a quienes se dedican a la edición.

Sin embargo, también se tiene que tomar en cuenta que algunos tecnicismos ya son imprescindibles para el medio editorial. Así como la jerga de los tipógrafos, impresores o diseñadores es de habitual conocimiento para quienes editamos, del mismo modo el lenguaje del desarrollador *web* o de *software* empieza a formar parte de nuestro bagaje cultural.

En esta primera entrada haremos una **comparación general entre algunos de los métodos más comunes para producir un libro electrónico estándar en formato EPUB**. En su momento se hablará con más detalle sobre la historia del EPUB.

Lo que ahora nos interesa es que, entre los distintos tipos de formatos existentes en el mercado del libro electrónico, el EPUB, desde sus inicios, fue pensado como un tipo de archivo para *ebooks* que sobresale por su **versatilidad, ligereza y seguimiento de estándares *web*** que aseguran una uniformidad en el código, y también un **completo control sobre la edición**.

Al tener estas características, el formato EPUB es fácilmente convertible a formatos privativos como los empleados por Amazon o Apple, asegurando así un ahorro de recursos y de tiempo al momento de crear una publicación digital.

Esta flexibilidad también ha permitido el desarrollo de diversos programas de cómputo que facilitan la creación de EPUB a cualquier persona. Con un par de clics desde el procesador de textos (Writer o Word) o maquetador (InDesign) instantáneamente tenemos un archivo EPUB a través de una conversión de formatos.

En principio, esto es una gran ventaja para autores independientes o para editoriales que no desean invertir «esfuerzos adicionales» en la producción de un libro electrónico. Sin embargo, existen al menos **dos desventajas** en estos procesos de publicación:

1. La limpieza del código, el control del diseño y la calidad de la edición en muchas ocasiones son inferiores a lo que es posible obtener a través de otros procesos.
2. Se pierde por completo de vista que lo más importante de la revolución digital dentro del mundo de la edición no es el libro electrónico, sino los procesos que ahora son posibles y que no representan algo «extra» al momento de producir un libro, sea impreso o digital. En su lugar, se trata de un conjunto orgánico de elementos que pueden ayudar tanto en el ámbito técnico de la publicación como en la calidad de la edición.

El libro electrónico es el punto de relieve más característico de la era digital en la edición, pero es solo la punta del *iceberg*. Para poder ir paulatinamente más a fondo tenemos que empezar a

familiarizarnos con **lo que está detrás de la producción de un *ebook***.

Como primera cuestión, existe cierto escepticismo sobre la necesidad de producir una publicación «desde cero». Es decir, se tiende a preferir el uso de conversores que, sin importar la estructura o el contenido de la publicación, automáticamente crean un archivo EPUB.

¿Para qué aprender lenguajes de etiquetado, como HTML o Markdown? ¿Para qué preocuparnos por las hojas de estilo, como CSS o SCSS? ¿Para qué reflexionar sobre el potencial de los lenguajes de programación para generar otras experiencias de lectura o para mejorar el cuidado de la edición, como JavaScript, Python, Ruby o C++?

Porque aunque el objetivo sea un objeto físico o digital, si empezamos a enfocarnos en su proceso de producción, poco a poco nos daremos cuenta de su pertinencia.

Características del ejercicio de comparación

Para mostrar las ventajas y desventajas que presentan los conversores de EPUB, en contraste con el desarrollo de una publicación digital «desde cero», se hará una comparación en la producción de una misma obra, pero con diferentes métodos.

Con la intención de que este ejercicio sea lo más «real» posible, se optó por tomar la edición del Proyecto Gutenberg del *Don Quijote*. Además, para mantener una uniformidad en la edición se ha partido del mismo texto en formato HTML y se ha usado la misma hoja de estilos CSS.

Algunas dudas respecto a estas decisiones pueden ser:

- **¿Por qué utilizamos la edición del Proyecto Gutenberg si hay mejores ediciones disponibles en línea como puede ser la publicada por el Instituto Cervantes?** La edición usada en este ejercicio es de dominio público, así como, a diferencia de la edición de Wikisource, fue sencillo descargar el libro entero en un solo archivo.
- **¿Por qué se parte de un texto previamente formateado, en lugar de vaciar el texto directamente desde los archivos descargados del Proyecto Gutenberg?** Al momento de hacer esta comparación se detectaron algunos descuidos en la edición que se decidieron enmendar (aunque sin duda se podrán encontrar más, ya que un cuidado estricto no es la meta de este ejercicio). Por otro lado, el formato del texto casi siempre se presenta como pesadilla para quienes editamos, provocando una mayor inversión de tiempo, recursos y esfuerzo únicamente para limpiarlo, cuestión que trataremos en otra entrada.
- **¿Por qué se emplea la misma hoja de estilo en lugar de recrear el diseño en cada uno de los métodos empleados?** El rediseño también involucra una mayor inversión de tiempo, recursos y esfuerzo, además de que el empleo de la misma hoja de estilo deja patente la pertinencia del diseño *web* para el mundo de la edición y la flexibilidad de poder utilizar esta hoja en los diversos métodos empleados (y obras), tema del que hablaremos en una futura entrada.
- **¿Cuáles son los métodos empleados en este ejercicio?** El libro se desarrolló de cuatro maneras distintas: desde InDesign, siendo la forma más común para la mayoría de los editores o diseñadores editoriales; con Jutoh, como ejemplo de *software* pensado para facilitar la creación de

publicaciones digitales; mediante Sigil, por ser uno de los programas más utilizados para la creación de EPUB, y «desde cero», lo cual involucra una intervención directa en los archivos necesarios para el desarrollo de un EPUB.

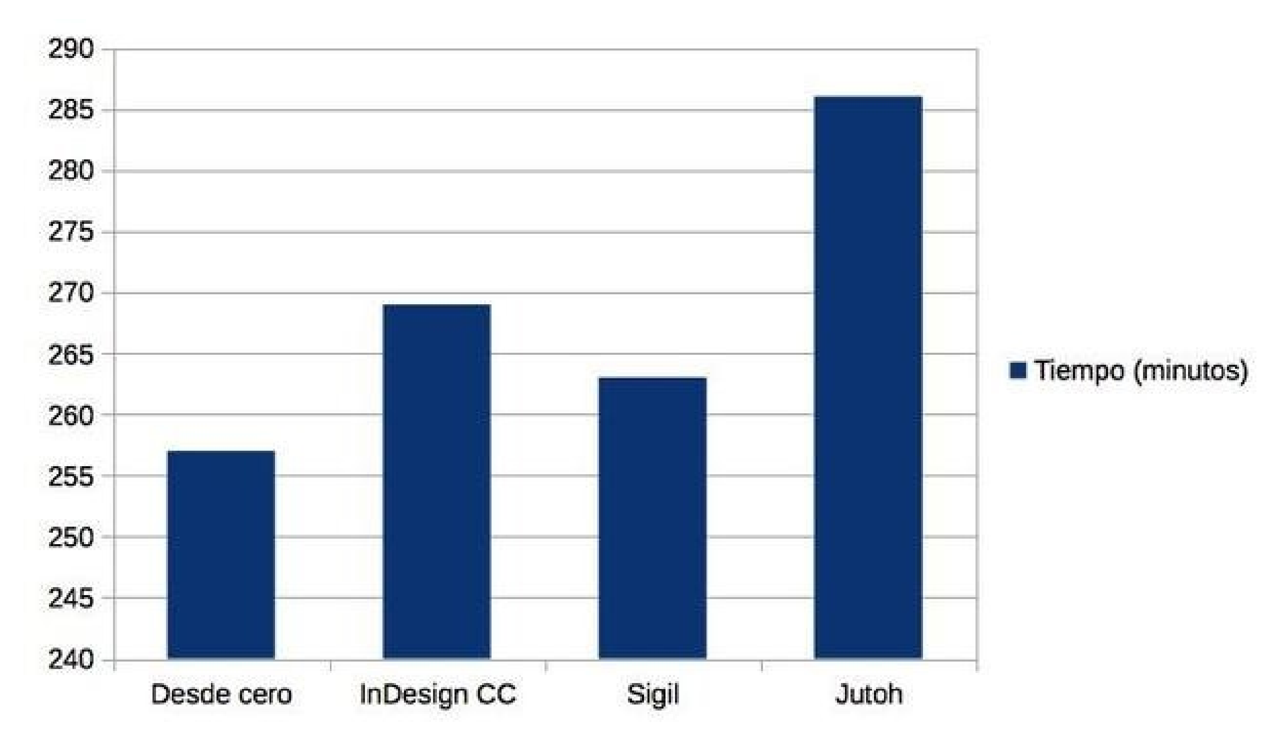
Comparativa de tiempos de producción: la eficacia del método «desde cero»

Uno de los mayores mitos que existe sobre la pertinencia de la creación de libros EPUB «desde cero» es que este modo de trabajar requiere mayores tiempos de producción. Sin embargo, el desarrollo «desde cero» no quiere decir que todo ha de hacerse de manera manual. Como lo estaremos viendo en otras entradas, mucho del trabajo al momento de desarrollar un EPUB es monótono y fácilmente automatizable a través de la creación de *scripts*.

Cuando hablamos de la creación de un libro EPUB «desde cero» nos referimos a que no existe un *software* intermediario que nos ayude a llevar a cabo esta tarea más allá de un editor de texto plano, o de código, y el uso de la línea de comandos.

En un primer momento esto puede sonar como un proceso demasiado complejo y que requiere mucha inversión de tiempo para aprender a llevarlo a cabo. Pero nada está más lejos de la realidad, si bien el desarrollo «desde cero» tiene sus retos, como cualquier otro método, es una habilidad que está al alcance de cualquiera que cuente con una computadora.

Si obviamos el tiempo empleado en darle formato al texto, en la siguiente gráfica podemos observar que en el desarrollo de la misma obra y diseño **el método «desde cero» es el más eficaz**. Esta diferencia se debe principalmente al tiempo que debe de emplearse en asociar las etiquetas HTML a los estilos creados con anterioridad.



Comparativa de tiempos de producción: la eficacia del método «desde cero».

Tanto en InDesign como en Jutoh es necesario asociar cada estilo CSS a estilos de párrafo o de caracteres; la diferencia entre ambos estriba en que en InDesign la asociación es más intuitiva y sencilla que en Jutoh. Por el otro lado, con Sigil o «desde cero» no existe la necesidad de asociación, ya que esta se detecta automáticamente al «enlazar» la hoja de estilos CSS a cada uno de los archivos del contenido del libro; sin embargo, el método «desde cero» presenta la ventaja de que no existe la necesidad de recrear el árbol de directorios o de importar archivos a un programa, como Sigil, para poder crear el archivo EPUB.

Tamaño del EPUB: la incidencia de las imágenes y el código «basura»

En cuanto al tamaño del archivo EPUB, existen al menos dos factores

que afectan directamente en su peso, que son **1)** las imágenes incrustadas, y **2)** la cantidad de código «basura».

La mayoría de los libros EPUB solo presentan una o dos imágenes para la portada o la contraportada, quizá una más si se incluye una fotografía del autor. Aunque se trate de un par de elementos, por lo general estas imágenes tienden a representar los **archivos con mayor peso en un EPUB** si existen alguna mezcla de estas circunstancias:

1. la publicación es relativamente breve,
2. las imágenes son más grandes de lo necesario,
3. las imágenes no tienen una buena compresión, o
4. las imágenes están en un formato poco pertinente.

Ninguna de estas posibilidades son las que afectan al peso de estos archivos EPUB debido a que todas están utilizando la misma imagen de 204 KB.

La diferencia de tamaños tiene una relación directa con la cantidad de código «basura». Cuando hablamos de la existencia de código «basura» queremos decir que, por los mismos resultados, algunos conversores añaden líneas adicionales de código. Sea para dejar patente el *software* que se utilizó para crear el archivo, o porque el método que asocia los estilos CSS con los estilos de párrafo o de caracteres es la recreación del archivo CSS bajo sus propios lineamientos y convención de nombres.

Esto produce al menos **dos desventajas**:

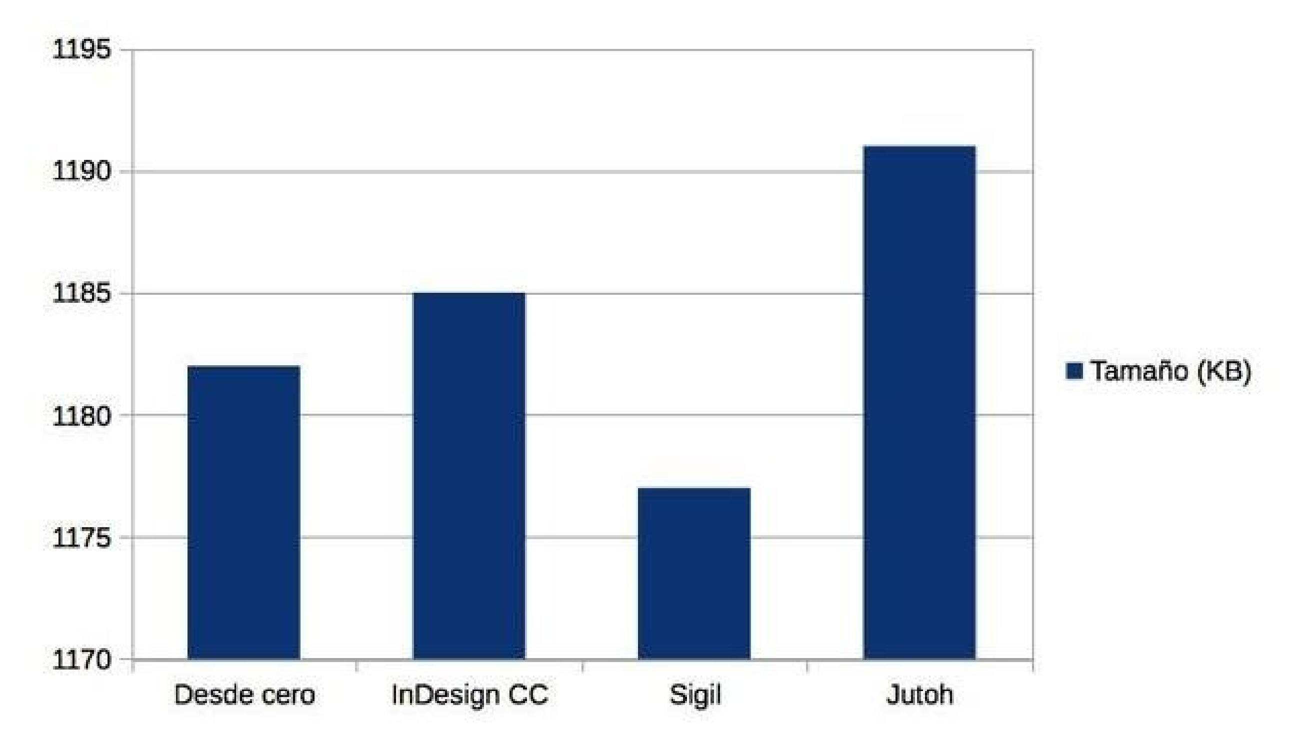
1. peso adicional e innecesario a la publicación, y
2. un cambio de nomenclatura que puede dificultar una ulterior modificación del diseño.

En el caso de los métodos con InDesign o Jutoh, el peso del libro se incrementó por el código «basura» existente; no obstante, la diferencia de peso entre los EPUB hechos con Sigil o «desde cero» es de otra índole que involucra la estructura interna del libro.

Sin extendernos demasiado, a partir de la tercera versión del formato EPUB existen dos archivos para la tabla de contenidos. El formato antiguo es el NCX, mientras que el formato introducido en la tercera revisión del EPUB es el XHTML o HTML.

Por defecto, Sigil solo te genera una tabla de contenido en formato NCX. Esto puede ser un potencial problema si consideramos que algunos de los lectores de libros más recientes solo soportan la visualización de tablas de contenidos en formatos XHTML o HTML.

Debido a esta circunstancia, **el EPUB creado «desde cero» contiene** tanto **una tabla de contenidos en formato NCX** (para lectores de libros antiguos) **y otra en XHTML**, lo cual representa 11 KB de peso adicional para una diferencia de solo 5 KB entre el libro desarrollado con Sigil y el creado con este método.



Tamaño del EPUB: la incidencia de las imágenes y el código «basura».

Errores y advertencias: verificación de los EPUB

Una de las principales ventajas que acarrea el proceso de producir un libro EPUB a través de conversores o *software* intermediario es que precisamente está pensado para usuarios con poco interés o conocimiento de HTML y CSS. La creación del libro, por lo general es de manera visual e incluso en algunos casos la curva de aprendizaje es relativamente corta.

Sin embargo, **además del cuidado editorial y del aspecto estético, un libro electrónico también tiene que tener una estructura interna coherente**. Es decir, si la finalidad es desarrollar un EPUB con calidad «profesional», este no debe de contener errores o advertencias frutos de inconsistencias en el lenguaje de etiquetado o de la hoja de estilos, en los metadatos, en los tipos de formatos aceptados o en la compresión de las imágenes.

Para que la búsqueda de inconsistencias sea efectiva y no nos demore mucho tiempo, existen los **verificadores de EPUB**. La herramienta oficial para verificar estos archivos se llama Epubcheck, la cual tiene una versión en línea como otra disponible para su descarga.

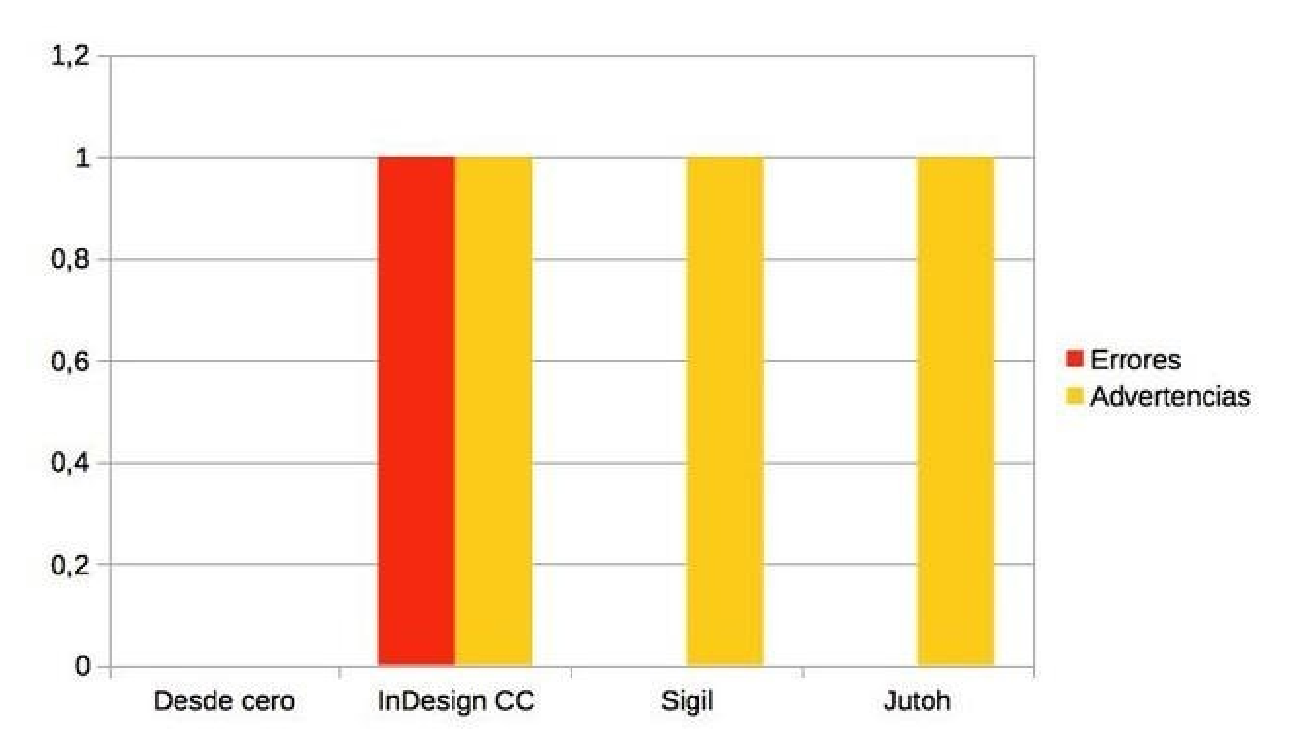
Sin embargo, por lo general un EPUB no solo se verifica mediante Epubcheck, sino con al menos otro *software*, sea por requerimiento del cliente, sea simplemente para tener otra herramienta de verificación. En el caso de nuestro ejercicio hemos utilizado tanto Epubcheck como una extensión para Firefox denominada BlueGriffon (verificador que varios de nuestros clientes piden utilizar).

Ahora bien, la gráfica que se muestra solo abarca los errores y advertencias detectados por BlueGriffon ya que ninguno de los EPUB presentó inconsistencias al ser verificados con Epubcheck. También cabe resaltar que la cantidad de errores o advertencias es poca debido a que se emplearon los mismos textos en formato XHTML y hojas de estilos, así como los metadatos fueron generados automáticamente por cada uno de los métodos. (La creación «desde cero» utilizó un *script* desarrollado inicialmente por el **colectivo Perro Triste** y que se ejecuta a través de la línea de comandos).

El error presente en el EPUB hecho con InDesign indica que este programa utilizó un tipo de compresión no válida para la imagen de la portada, mientras que la advertencia es exactamente la misma a las que detecta en los EPUB creados con Jutoh y Sigil: un elemento presente en los metadatos se considera obsoleto.

En realidad, **resolver estos errores y advertencias no es nada complicado;** no obstante, para un usuario que no está familiarizado con la estructura del EPUB el proceso para enmendarlos puede ser frustrante. Para arreglar estos EPUB es

necesario su descompresión de manera externa para ir directamente al archivo que presenta el problema, modificarlo y, por último, volver a comprimir el EPUB.



Errores y advertencias: verificación de los EPUB.

Costos implícitos en la producción: programas de pago versus *software* libre

Al momento de producir un EPUB, por lo general el usuario se pregunta si los programas de cómputo necesarios son gratuitos o si tienen algún costo. Afortunadamente, para la producción de un libro electrónico no es necesario pagar por alguna licencia de *software*.

En la mitad de los cuatro métodos vistos en este ejercicio se empleó *software* privativo que implica costos para su uso. Tanto InDesign como Jutoh requieren el pago de una licencia, mientras

que Sigil es *software* libre. Para el método «desde cero» se emplearon programas de *software* libre (Atom, Firefox y el *script* desarrollado por Perro Triste) y del sistema (la terminal).

Un mito recurrente entre los usuarios que no están familiarizados con el uso de *software* libre o de código abierto es que su gratuidad es sinónimo de baja calidad. La falsedad de este supuesto —al menos en lo que concierne a la producción de libros EPUB— pudo observarse en este ejercicio: **Sigil y el método «desde cero» arrojaron mejores resultados**.

No obstante, tampoco puede dejarse de lado que la gran mayoría de las editoriales emplean InDesign para la edición de sus libros impresos, por lo que en determinadas circunstancias lo más conveniente quizá sea irse por esta misma vía para desarrollar libros electrónicos.

Si esta no es tu situación y, al mismo tiempo, deseas mejorar la calidad de tus EPUB, por lo que estás considerando hacer una inversión en *software* privativo, mejor piénsalo dos veces. Existen otras posibilidades libres y de alta calidad que pueden acomodarse a tus necesidades.

Conclusión: el método «desde cero» gana esta partida

A lo largo de este ejercicio quedó evidenciado que uno de los métodos más eficaces y eficientes para la producción de libros electrónicos estándar es el **desarrollo «desde cero»**. Varios lectores podrán pensar que este modo de producción implica ciertos conocimientos técnicos complejos y una mediana o larga curva de aprendizaje.

Como experiencia, puedo compartirles que a las personas que, por medio de Nieve de Chamoy o Perro Triste, les hemos ayudado a

aprender a hacer EPUB, no les ha tomado más de 24 horas de taller el desarrollo de su primer libro electrónico «desde cero». Además, cabe la pena resaltar que la mayoría de estas personas provienen de un contexto de completo desconocimiento de los lenguajes HTML y CSS, así como del uso de la línea de comandos.

Sin embargo, como tampoco se trata de reducir las posibilidades de producción a este único método, **en futuras entradas veremos cómo elaborar libros EPUB desde InDesign, Sigil y el desarrollo «desde cero»**. No trataremos el método desde otros programas (como Jutoh o iBooks Author) porque no lo considero conveniente.

Si se tiene planeado el uso de *software* exclusivamente para crear libros electrónicos, la recomendación es que sea *software* libre o de código abierto, para que no involucre costos de producción extra, y cuyos formatos de salida sean estandarizados, como el EPUB (el cual sin problemas puede convertirse a formatos privativos como los usados por Amazon y Apple); por ello, **la recomendación bajo estas necesidades es Sigil**.

Por otro lado, para no obviar la tendencia editorial de **producir libros con InDesign**, también se verá **la manera más óptima de desarrollar libros EPUB con este programa**. Pero antes de esto, el siguiente tema a tratar será sobre un problema rutinario en el quehacer editorial y que, aunque parezca mínimo, representa uno de los cuellos de botella al momento de producir libros, sean impresos o digitales; es decir, hablaremos sobre **la manera más adecuada de dar formato a un texto**.

Sobre esta entrada

- Título original: «Edición digital como edición "desde cero"».
- Título en el *blog* de Mariana: «Edición digital como edición

“desde cero”».

- Fecha de publicación: 6 de septiembre del 2016.
- Redacción original.
- Revisión de Mariana.

Edición cíclica y edición ramificada: de la antesala a la «revolución» digital

De la antesala a la «revolución» digital

La edición de material impreso nunca ha sido un proceso sencillo. Desde los amanuenses hasta la imprenta se han requerido diversas habilidades para poder cumplir con cada uno de los procesos. Los procesos para publicar una obra, por lo general, son:

1. Editar y corregir.
2. Diseñar y maquetar.
3. Cotejar y revisar.
4. Imprimir.
5. Distribuir.
6. Comercializar y administrar.

Antes de la «revolución» digital, la minuciosidad puesta en cada proceso marcaba el ritmo en el flujo de la edición. Si se descuidaba un elemento, con seguridad la obra perdería calidad o tendría que empezarse de nuevo. Por este motivo, **cada proceso de la edición tradicional iba paso a paso, tratando de regresar lo menos posible a la fase anterior**.

Este andar también iba al ritmo de la infraestructura, los recursos y la demanda existente. Antes de la imprenta el flujo era consistente y con un «lento» crecimiento. Con la llegada de la impresión la edición se aceleró, los costos decrecieron y la demanda creció.

El libro pasó de ser un artículo poco asequible a ser un elemento de uso diario. Podemos decir que la imprenta no solo fue una «revolución» sino también una «apertura» de la información.

El traslado a lo mecánico permitió que los oficios relacionados con la edición se especializaran a través de los siglos. Si bien las técnicas se fueron afinando o cambiando hasta mediados del siglo xx la concepción metodológica permaneció invariable: **la edición era una progresión unilateral hasta la publicación y comercialización de la obra**.

En la «revolución» digital

El uso de tecnologías analógicas implicaba que solo unos pocos podían solventar algún revés. Si una publicación salía con defectos, no quedaba más que la adición de una fe de erratas o, en el peor de los casos, la bancarrota.

A partir de la segunda mitad del siglo xx esta condición empezó a flexibilizarse. La masificación de las computadoras abrió la puerta para facilitar la regresión en los procesos, al permitir la modificación del documento a diestra y siniestra, incluso durante su impresión.

La publicidad era clara durante el surgimiento digital: *la máquina de escribir en la oficina* siempre dejaba rastro en la corrección. Este fue el primer frente en el que las computadoras personales buscaron un mercado, pero fue cuestión de tiempo que observaran otras posibilidades:

- las computadoras en lugar de *la máquina de escribir en el hogar*, para la redacción de documentos personales o domésticos;

- la tecnología digital para «modernizar» tanto la escritura como la impresión de documentos.

Las computadoras dividieron al mundo editorial. La confrontación no fue fortuita: el estado de la tecnología digital representaba un revés en la calidad editorial alcanzada a través de siglos. Pero paulatinamente se fue puliendo (¿o ignorando?) la calidad de la edición digital gracias a TEX y el DTP (*desktop publishing*).

Algunos elementos de la tradición editorial fueron rezagados. Sin embargo, **la concepción metodológica de una progresión unilateral permaneció inalterable**. La diferencia estribó en que la tecnología digital permitió que el retorno en los procesos se hicieran por medio de un teclado y un ratón. Al menos así fue cómo las compañías de *software* y *hardware* publicitaron la apuesta por la «modernización» en la edición de documentos.

En los noventa, el DTP como sinónimo de edición estaba en proceso de consolidación. La edición era ya edición digital. De Corel Ventura, pasando por PageMaker y QuarkXPress, hasta InDesign, la publicación de *impresos* se ha vuelto un proceso cada vez más centralizado en un paquete de *software*.

Junto a la «revolución» digital en los procesos de edición también vino otra «apertura» en la información: **el libro dejaría ser sinónimo de impreso**. El sueño de un libro sin páginas se fue concretando en un archivo digital. Del papel al *byte*, los libros electrónicos surgieron como una alternativa.

El recelo ante otros soportes distintos al impreso aún sigue siendo una actitud muy común entre los editores. De nuevo la discusión se centra en la calidad del soporte: todavía en la actualidad el libro electrónico carece de los estándares de calidad según el parámetro del impreso.

En lugar de apoyar al mejoramiento de los formatos digitales **el grueso del mundo editorial ha decidido apostar por la cautela y la espera de que agentes externos al sector mejoren la calidad editorial del *ebook***.

Varios editores han decidido abrir la puerta al libro electrónico, en muchos casos más como una cuestión de «adaptación a los tiempos» que un sincero anhelo por diversificar los soportes de una obra. La discusión se traslada a una cuestión de costos: producir más formatos implica una inversión proporcional de tiempo y recursos según la cantidad de formatos «adicionales».

Para lidiar con la necesidad de múltiples formatos se emplea una metodología que consiste en terminar la publicación del impreso para después pasar a la elaboración de otros formatos. Esto provoca un aumento en el tiempo de publicación, según la cantidad de soportes, y también la idea de que los procesos de producción de un libro electrónico son de distinta naturaleza a la producción de un impreso. Asimismo, se ignora que en esta sucesión se heredan tanto contenidos como erratas y elementos innecesarios para el resto de los formatos.

La progresión unilateral y su posibilidad de regresión —característica de la edición tradicional— mutan en una serie de ciclos que brincan de un formato a otro. **La metodología cíclica es la pretensión de mantener un desarrollo unilateral cuando el objetivo es la consecución de diversos soportes**. Es decir, en la edición tradicional se iba paso a paso hasta llegar al impreso; ahora se continúa con esa idea pero implica dar más pasos para tener más archivos.

El cambio de método queda invisibilizado al entenderse cada nuevo ciclo de producción como *procesos adicionales* a la publicación del impreso, que comúnmente se considera el soporte

«nuclear» del resto de los formatos.

Que los editores piensen que los procesos necesarios para la publicación en otras versiones distintas a la impresa no son ni complementarios ni distintos es, y siempre ha sido, el meollo en la discusión sobre el supuesto antagonismo entre las publicaciones impresas y las electrónicas.

Más que una cuestión de calidad o de costos el escepticismo del mundo editorial gira en torno al carácter mítico de la inmutabilidad metodológica y a que existe la necesidad de una adopción consciente de un método que no sea unilateral.

Me explico: la supuesta confrontación entre las versiones impresas y las digitales, y el escepticismo ante los formatos digitales, no es porque estos sean de menor calidad o porque el costo de producción no justifique su «inversión», sino a la idea o mito de que la producción de un libro discurre en un solo camino, una sola línea, paso a paso, y la obstinación de no cambiar de método.

La discusión y la confrontación es esencialmente metodológica, no de técnica (porque las técnicas han variado a través de los siglos), no de formatos (que no están reñidos entre sí porque hay lectores con distintas preferencias), no de calidad (que se mejora con el tiempo, como la edición digital para PDF de impresión) y tampoco de costos (ya que con una metodología correcta el costo sería mínimo).

Mientras se continúe pensando que los ciclos son soluciones *ad hoc*, en lugar de concebirse como un método, la mayoría de los editores solo entenderán la «revolución» digital como una «revolución» en formatos y técnicas, y no como algo más profundo: un cambio y «apertura» en la forma de «publicar» cultura.

Por estos motivos, aunque al parecer exista una continuidad, **la**

edición cíclica es metodológicamente distinta a la edición tradicional. La edición cíclica aborda el desafío traído por el multiformato como una presunta continuidad metodológica de la edición tradicional, pero añadiendo ciertos «módulos» para la producción de otros formatos.

La edición cíclica se concentra en producir varios soportes al mismo tiempo que quiere conservar la idea que la publicación de un libro es un *solo* camino a seguir. Esta concepción metodológica está tan arraigada que el «*software* milagro» da tranquilidad al concebir la conversión entre formatos como la solución final. Solo cuando se percibe la pérdida en la calidad editorial y técnica se hace patente que la edición cíclica no es el método más óptimo para la producción multiformato…

Sobre esta entrada

- Título original: «Edición cíclica y edición ramificada».
- Título en el *blog* de Mariana: «Edición cíclica y edición ramificada: de la antesala a la "revolución" digital (1/3)».
- Fecha de publicación: 4 de julio del 2017.
- Redacción original.
- Revisión de Mariana.

Edición cíclica y edición ramificada: del surgimiento al tronco y las ramas

Surgimiento de una idea

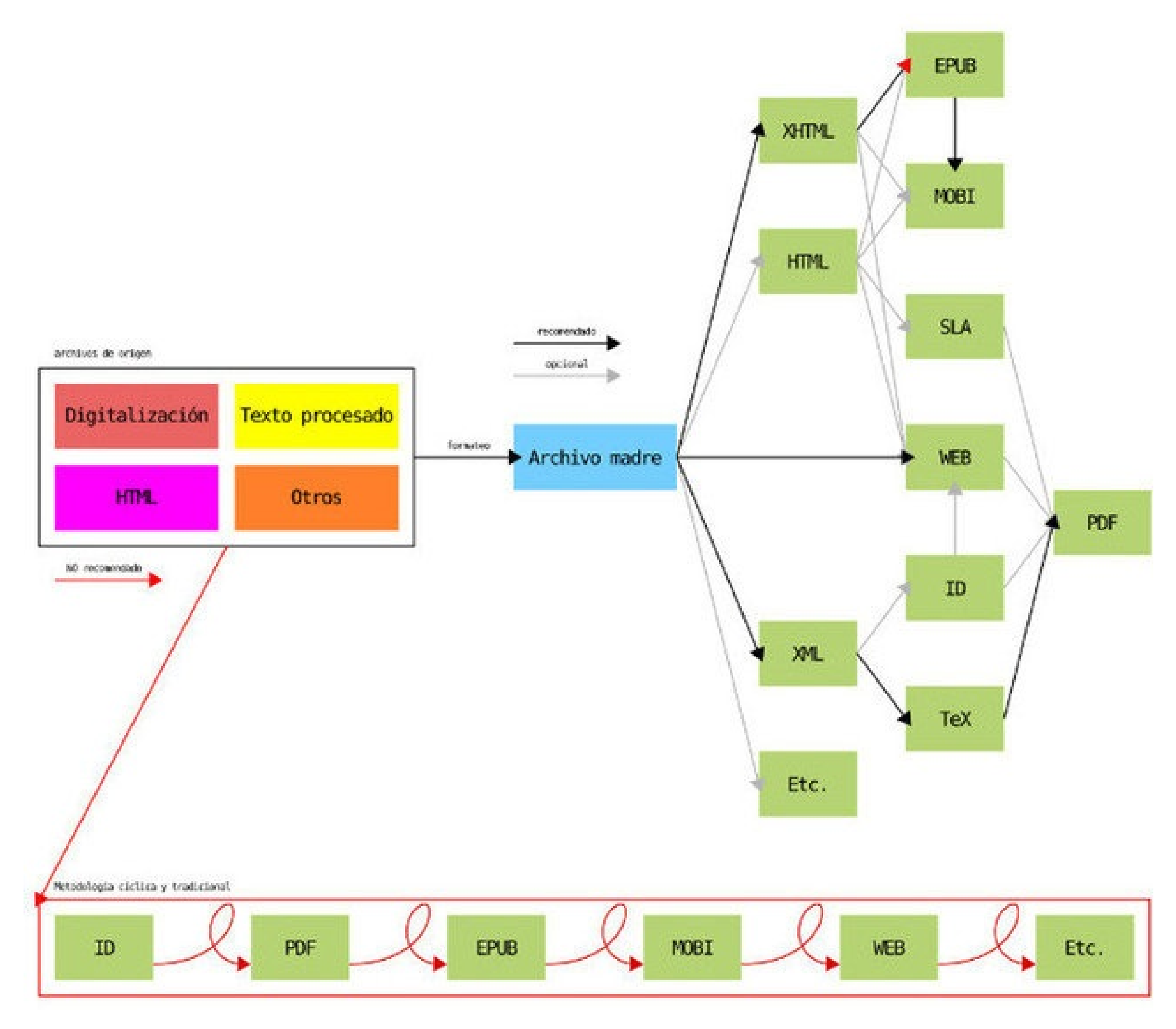
¿Y si cambiamos de perspectiva? ¿Si en lugar de hablar sobre la diversidad en los formatos finales, pensamos en un método multilateral? La idea es sencilla: **no nos concentremos en los formatos, sino en los caminos que llevan a ellos**. El supuesto también es evidente: a múltiples formatos, diversos senderos.

Pese a que la idea es sencilla, en el grueso editorial se interpretará como un discurso que apuesta por la pretendida destrucción de su tradición, cuando en realidad atenta a un ideal metodológico, no a los conocimientos que a lo largo de siglos se han ido acumulando para publicar «buenos» libros.

Una prueba de esta extrapolación es patente cuando se analiza la tradición editorial: las técnicas han variado a través de los siglos, pero la idea de ir a un punto A a uno B ha permanecido.

En la edición tradicional el punto A era igual al texto original y el punto B igual al impreso. En la edición cíclica A permanece, mientras que B es el fin de todos los ciclos, obteniéndose así no solo el impreso, sino también al menos un formato digital.

Sin embargo, en la **edición ramificada** no hay una consecución de A a B, sino **un tronco (A) con diversas ramas (B, C, D…) que cesan su crecimiento, otras se desprenden y unas más se convierten en formato para más ramas**.



Esquema comparativo entre la edición cíclica y la ramificada.

Esta concepción metodológica surgió de un campo especializado: la elaboración de documentación de *software*. En el surgimiento de la era digital pronto se vio la necesidad de textos que explicaran el uso del *software*. Sin embargo, debido a las distintas preferencias de consulta y de *hardware*, se hizo menester crear la documentación en distintos formatos.

Pronto se percibió la dificultad presente en la edición cíclica: cada nuevo soporte implica un aumento en la inversión de tiempo

y de cuidados, y por ende, de recursos. Para combatir este problema y ahorrar en material impreso, a partir de los noventa surge la idea del *single source publishing* (SSP), cuya característica es el desarrollo multilateral de diversos formatos.

Aunque la idea tiene poco más de veinte años, es ahora cuando empieza a expandirse para el resto del quehacer editorial. En la actualidad, la edición ya no solo es digital, sino también la publicación. **Los libros electrónicos han puesto en el centro de la discusión el problema metodológico que el sector editorial viene arrastrando desde el inicio de la «revolución» digital**.

El tronco y las ramas

Pero, en concreto, ¿qué es la edición ramificada? Lo primero que podría entenderse es que significa una apertura en todo sentido. No obstante, el término «*single source*» ayuda a esclarecer el **primer elemento** de esta metodología de trabajo: **la edición empieza con un «archivo madre»**.

El «archivo madre» es un documento a partir del cual se crean el resto de los formatos: es el tronco. Este archivo puede ser tanto el texto original del autor como el documento editado, ya que el elemento mínimo necesario es que esté en un lenguaje de marcado.

El archivo madre obedece a una dimensión estructural a partir de un conjunto de etiquetas que indican cada elemento del texto (véase aquí para más información). La edición del texto puede darse antes o después de esta estructuración.

Lo recomendado es que la edición y la estructuración se den en conjunto, porque antes de la publicación quien edita es uno de los principales conocedores de la obra. La desventaja es que se requiere saber al menos un lenguaje de marcado, pero esto se solventa si se

recurre a un lenguaje de marcas ligero.

Los lenguajes de marcas ligeros son cómodos de leer, fáciles de escribir y sencillos de convertir. Por este motivo, es preferible un lenguaje «ligero», como Markdown, que uno «completo», como XML, HTML o TeX. (Véase la entrada original de este artículo como ejemplo).

El **segundo elemento** de este método es ir de lo simple a lo complejo. Cada formato presenta sus propias particularidades. En el impreso se requieren ajustes manuales por cuestiones ortotipográficas o de diseño; en el EPUB a veces es preciso ordenar gráficas o tablas para una mejor visualización; en la lectura *online* sin inconvenientes se saca provecho de la visualización interactiva de información, etcétera. Para evitar la herencia de características, como sucede en la metodología cíclica, la edición ramificada parte de un documento con los elementos comunes, para después hacer ajustes según el caso.

El traslado del archivo madre a cada uno de los formatos finales exige, en la mayoría de los casos, el uso de un nuevo lenguaje. Si se parte de Markdown, se requiere XHTML para un EPUB, XML para InDesign o TeX para LaTeX o ConTeXt. Si bien se recomienda el conocimiento de estos lenguajes, el **tercer elemento** es el uso de conversores para el ahorro de tiempo.

El trabajo con un lenguaje de marcado posibilita la automatización de la traducción a otros lenguajes. Sin embargo, la traducción realizada por una máquina, por lo general, necesita modificaciones y cotejos. Los conversores no son una solución final, pero evitan el trabajo monótono.

El mejor *software* que puede ayudarnos en esta tarea es **Pandoc**. Este programa libre es de los más poderosos que podemos encontrar. Y, si bien en algunos casos el resultado no es el esperado,

el tiempo involucrado en los ajustes es menor a volver a formatear el documento.

Esta inversión de tiempo permite diferenciar la edición ramificada de muchos *software* milagro que se ofrecen en el mercado. Esta clase de programas se autoperciben como la «solución» a varios problemas que el editor encuentra al momento de publicar en múltiples soportes.

La edición ramificada no es un *software* ni un lenguaje, mucho menos una solución y tampoco un entorno de trabajo —guiño a Adobe—. **La edición ramificada es una metodología que puede manifestarse de múltiples maneras**, unas más acabadas que otras. Al ser un método también cuenta con la posibilidad de pulirse o de descartarse si crea inconvenientes.

Por ello, el **cuarto elemento** es que la edición ramificada solo está pensada cuando existe la posibilidad de múltiples formatos. Si la obra se proyecta como un «libro objeto» o una publicación artesanal basada en métodos analógicos este método no tiene cabida.

Los conversores no son lo único que permiten el ahorro de tiempo. El **quinto elemento** consiste en que el tamaño del equipo de trabajo es proporcional a la agilización y división del trabajo. Una diferencia nítida entre la edición cíclica y la ramificada es que en la última ningún formato final parte de otro, lo que permite el trabajo paralelo en la producción de cada formato.

Cada soporte emprende su camino de manera simultánea a partir del archivo madre, resaltando el **sexto elemento** metodológico: la edición ramificada es edición sincrónica, independiente y descentralizada.

Uno de los temores que surgen en la edición ramificada es que la simultaneidad y autosuficiencia puede dar cabida a una

divergencia en el contenido, ya que no existe un mecanismo centralizado para el control de la edición. No obstante, en muchos casos la edición cíclica implica una gran pérdida de control, debido a que el encargado del cuidado editorial tiende a desconocer los procesos «periféricos» y «adicionales» a la producción del soporte impreso.

Si hablamos de «control», en el contexto digital **el cuidado de una obra no solo debe de ser editorial, sino también técnico**. Si no hay dominio sobre el código no existe la seguridad de que el texto, aún con lenguaje de marcado, sea fácil de convertir o sencillo de leer y analizar.

La pérdida de control puede resolverse de tres maneras distintas:

1. Realizar la edición y las estructura de la obra al mismo tiempo, llevando a cabo ambos procesos de manera simultánea.
2. Borrar el archivo creado y volver a generarlo a partir del archivo madre; la vía más cómoda pero que supone volver a pulir los detalles derivados de la conversión.
3. Crear un programa para que analice cada archivo de cada formato y lo compare con el contenido del archivo madre. De este modo, si en un formato A la palabra «andará» accidentalmente se cambia por «andara», el *software* lanzaría una advertencia de que en el archivo madre esa palabra es «andará». Esto permitiría encontrar modificaciones accidentales del texto cuando se esté modificando la estructura o el diseño de manera manual. Y sería solo una advertencia, ya que el cambio podría ser intencional, como «murcié\-lago» en TeX a comparación de

«murciélago» en el madre, solo para sugerir una separación silábica para arreglar una caja. Para esto es necesario obtener las palabras del texto, ignorando la sintaxis de cada lenguaje de marcado.

No obstante, en más de una ocasión existen correcciones de último momento que también han de añadirse a cada uno de los formatos. La corrección manual involucra más cotejos e incluso abre la puerta a más erratas. El **séptimo elemento** llama al uso de diccionarios y de expresiones regulares (*regex*) para evitar o automatizar las modificaciones. El diccionario permite detectar erratas en el archivo madre. El uso de *regex* facilita corregir los archivos sin la necesidad de ir caso por caso.

El uso del diccionario es recomendable únicamente para el cotejo de posibles erratas, sea caso por caso o mediante la creación de una lista ordenada con las palabras dudosas. En cuanto a *regex*, hay que tener cautela con su uso. El dominio de las expresiones regulares se adquiere con el tiempo, por lo que las primeras implementaciones tienen que ser básicas y después de hacer sido aplicadas en pruebas.

Continuando con la idea de un *software* que monitoree las ramas, este también tiene que permitir la corrección automatizada. El procedimiento sería la realización de una corrección manual en el archivo madre que, de manera automática, el programa traslade los mismos cambios a cada formato.

Sobre esta entrada

- Título original: «Edición cíclica y edición ramificada».
- Título en el *blog* de Mariana: «Edición cíclica y edición

ramificada: del surgimiento al tronco y las ramas (2/3)».

- Fecha de publicación: 11 de julio del 2017.
- Redacción original.
- Revisión de Mariana.

Un vuelo seguro y constante

Hasta aquí se ha descrito cómo es posible la producción de diversos formatos y posteriores modificaciones sin dependencia a un soporte final. Pero hay ocasiones en que el proyecto completo se arruina: archivos corruptos, pérdida de documentos, malfuncionamiento de la computadora, etcétera.

La generación de respaldos, entendidos como copias adicionales almacenadas físicamente o en la «nube», se vuelve un requisito indispensable para los proyectos de cualquier índole. De esta manera, cuando aparece un fallo se retoma alguna de las copias.

Sin embargo, esto acarrea los siguientes inconvenientes:

- Mayor necesidad de espacio cada vez que se crea una copia.
- Posibilidad de confusión entre los respaldos y el proyecto actual.
- Falta de control en los cambios, principalmente en grandes equipos.
- Dificultades al trabajar de manera separada o remota.
- Desconocimiento de casi todas las modificaciones que se han llevado a cabo.

El descuido en la información afecta la calidad de la edición. Por lo que el mantenimiento de la información no solo es una tarea secundaria o «técnica», sino una responsabilidad editorial.

El **octavo elemento** metodológico es el control de versiones. En

un videojuego existe la posibilidad de guardar la partida a cada paso o en algunas situaciones críticas. A partir de estos cambios guardados es posible regresar o probar nuevos caminos. Si el desarrollo es satisfactorio estos otros senderos se convierten en el juego principal.

Un control de versiones cuenta con esos elementos. A partir de un programa es posible monitorear las carpetas y archivos de un proyecto. Se hace posible tener conocimiento de todo lo hecho, sin posibilidades de confusión ni necesidad de mayor espacio, porque se emplean puntos guardado.

En un videojuego no se respalda el juego una y otra vez para guardar una partida, solo se almacenan los datos necesarios para regresar a ese punto de guardado. **En un controlador de versiones no hay respaldos, sino información para volver a cada punto de control**.

Esto implica que el espacio necesario para guardar un proyecto cae drásticamente. También resuelve la dificultad de trabajar de manera independiente, ya en el tronco como en las ramas e incluso en sus partes pueden colocarse puntos de control en cualquier momento. Así, se puede saber si se está trabajando con versiones anteriores o si hay sobrescritura, porque cada punto de control tiene descripción, fecha, autor, identificador e indicador de qué se creó, modificó o eliminó.

En un videojuego el uso de partidas guardadas permiten explorar más a fondo la trama del juego debido a que en cualquier momento es posible regresar. En un controlador de versiones hay tal flexibilidad que es posible experimentar ciertas implementaciones (como *regex*, p. ej.) sin el peligro de corromper un proyecto. Esto se traduce a que yace la posibilidad de adquirir mayores conocimientos de edición digital sin el temor de

arruinarlo todo.

Pero no solo eso. Un proyecto con un control de versiones es lo que se conoce como repositorio y estos tanto pueden estar en una computadora como en la «nube». El **noveno elemento** es alojar los proyectos en un servidor para que cualquier colaborador, desde cualquier lado y en cualquier momento, pueda «subir» o «bajar» información para realizar su trabajo.

El SSP pasa a ser SSOP (*single source and online publishing*). El SSOP se utiliza de manera muy extensiva para el desarrollo de *software* o investigaciones científicas, pero también cabe emplearse en el quehacer editorial.

La edición ramificada es una expresión particular del SSOP. El nexo es importante ya que el SSOP se enfoca a un control en los procesos, principalmente técnicos. Mientras tanto, los elementos metodológicos de la edición ramificada hacen que este control se traduzca en un mayor cuidado editorial, donde las tecnologías digitales y la automatización de procesos sirven de apoyo a quien edita.

La «revolución» digital no va en contra de los conocimientos adquiridos durante siglos, sino que revisita, depura o elimina técnicas y metodologías que no son las más pertinentes para llevar a cabo una tarea. En la edición, la «revolución» digital es la «apertura» del «gremio» al uso de las tecnologías de la información en pos de técnicas más aptas pero sin olvidar lo que hace de una obra un «buen» libro.[1]

Sobre esta entrada

- Título original: «Edición cíclica y edición ramificada».

- Título en el *blog* de Mariana: «Edición cíclica y edición ramificada: un vuelo seguro y constante (3/3)».
- Fecha de publicación: 18 de julio del 2017.
- Redacción original.
- Revisión de Mariana.

¿*Single source* y *online publishing*, fuera de la realidad?

Para continuar con nuestro receso técnico, quiero hablar sobre el título de esta entrada: **¿*Single source* y *online publishing*, fuera de la realidad?** Mi principal interés es seguir la discusión de los argumentos aquí propuestos en los comentarios (y espero que esto no merme el deseo de publicarlos).

Como hemos visto a lo largo de diversas entradas, el método que se ha propuesto como el más afín para satisfacer las necesidades editoriales actuales es el *single source* y *online publishing* (expresiones que trataremos como una entidad única, y cuyo acrónimo es SSOP).

Por medio de esta metodología se pretende **atajar el «problema» que implica la producción de diversos formatos para una misma obra**, así como la **pérdida de control en la edición** al momento de usar una computadora. (Para mayor detalle, lee «Historia de la edición digital)».

Como toda propuesta, esta no está ausente de flaquezas y detractores. Por tanto, me gustaría hacer algunas puntualizaciones y, ojalá, provocar que tú también des tu opinión.

1. El *single source* y *online publishing* es demasiado técnico para el editor

Cuando se intenta explicar el uso de TeX al medio editorial más de un editor lo considera tecnológicamente complicado. TeX es percibido como no apto para los tiempos y los requerimientos que

manejan en sus libros. Algo semejante sucede con el *single source* y *online publishing*: a los editores no les parece que sea técnicamente viable, ya que implica un cambio en la manera de editar sus libros.

Mónica Braun, mi colega de Nieve de Chamoy —y quien cuenta con una experiencia editorial que sobrepasa mi edad (¡pero no por mucho!)—, es una de las más severas críticas de este método. Más de una vez ha sido el motivo de discusión para editar y publicar los libros de Nieve de Chamoy. El uso de este método —opina junto con otros— no está al alcance del editor que está habituado (*¿mal acostumbrado?*) al uso del *software* que le provee Adobe o Microsoft.

Ellos están en lo correcto. El estado actual de esta metodología no es lo suficientemente madura como para que en el presente reemplace a los procesos tradicionales de producción editorial. A pesar de ello, quienes pueden ya implementar este método empiezan a observar su pertinencia.

Por ejemplo, el primer libro de Nieve de Chamoy, *80 años: las batallas culturales del Fondo*, con más de 150.000 palabras, 378 notas al pie y aproximadamente 70 fuentes bibliográficas, nos tomó una semana producirlo en EPUB, y porque lo hicimos a marchas forzadas; dos o hasta tres semanas si hubiéramos tenido el tiempo suficiente.

Actualmente, con el SSOP y con las herramientas que en Perro Triste hemos estado creando y afinando, la producción de este libro **nos tomaría solo una jornada de trabajo**. No solo eso, la obra tendría una **edición más cuidada, un mejor aspecto estético y con salida para diversos formatos**. (Actualmente, estamos trabajando en la segunda versión de los libros de Nieve de Chamoy).

Además, se suele olvidar que en los procesos editoriales no solo el editor está involucrado. El editor promedio, por lo general, nunca absorbe todos los procesos y, *principalmente*, no tiene que hacerlo.

Este método para mantener el control en la edición, sin importar la diversidad de formatos así como para agilizar los tiempos de producción, no necesariamente los tiene que aprender el editor, sino que está orientado a otros perfiles profesionales, como los del diseñador editorial, el tipógrafo o el desarrollador.

2. El *single source* y *online publishing* no está pensado para los editores

Sí y no. El *single source* y *online publishing* tiene su principal enfoque en afrontar una realidad que ha sobrepasado a la edición tradicional: la **publicación de diversos formatos con el mismo cuidado y calidad editorial sin que esto aumente proporcionalmente la necesidad de recursos y de tiempo**.

Los procesos tradicionales de edición tienden a publicar formatos por ciclos, muy a tono con la estrategia y la capacidad de la imprenta. Y esa manera de trabajar ha sido trasplantada a las recientes necesidades de la publicación digital.

La contradicción está anunciada: **¿por qué se inicia un proceso digital** (la edición, el diseño e incluso la impresión se llevan a cabo *a través de computadoras*) **para obtener un producto físico y luego *regresar* para producir un producto digital?**

Como puede observarse, este problema no se origina por el quehacer tradicional del editor. La corrección, la verificación, la inspección, el cuidado ortotipográfico y más, no se ven amenazados por esos procesos *subsiguientes*. Sin embargo, sí pueden llegar a obstaculizarlos si el editor emplea formatos no adecuados (y que nunca fueron pensados) para el cuidado editorial.

Si hacemos caso omiso a esta posibilidad, efectivamente el *single source* y *online publishing* viene para cambiar el paradigma

cíclico y, en su lugar, implementar un **modelo ramificado**. El texto editado es al *single source* y *online publishing* lo que el tronco es al árbol. **El contenido editado es el elemento madre desde el cual se obtienen diversos formatos y a partir de él se bifurcan distintas ramas**; es decir, se obtienen distintos formatos de salida (EPUB, HTML, MOBI, PDF, etc.).

Este proceso se realiza de manera simultánea y con mayor independencia entre una y otra rama. Se evita que esta bifurcación se produzca una vez que un formato ha sido concluido y se elude la herencia de las características (léase errores) del formato previo.

¿Peligro a la pérdida de control? Solo si el editor continúa dándole preferencia al aspecto visual, en lugar de preocuparse por el **marcaje semántico del contenido**. Además, con un controlador de versiones, como los implementados por los repositorios, siempre es posible regresar a puntos de guardado anteriores, sin necesidad de preocuparnos por el típico respaldo, del respaldo, del respaldo.

En fin, el *single source* y *online publishing* si bien se orienta a esta bifucación posterior a la edición habitual, requiere que el editor adopte este enfoque semántico. Por tanto, sí está pensado para su trabajo y, esencialmente, para **evitar la reproducción de vicios al momento de trabajar con archivos digitales**. De nueva cuenta, Mónica es un buen ejemplo de cómo un editor, al momento que deja el enfoque visual y adopta la perspectiva semántica, mejora el cuidado de su trabajo.

Cuando utilizábamos directamente los archivos del procesador de texto se nos iban muchas horas en limpiar ese formato y etiquetarlo. Pero, principalmente, en ajustarlo, debido a diferentes criterios en la estructura que impactaban en el aspecto visual del texto o, por ejemplo, en revisar que no se hubiera escapado ni una sola itálica en toda la obra.

Ahora Mónica trabaja directamente con Markdown y así se encarga de la cuestión semántica. Esto ha hecho que el tiempo de producción haya disminuido. Una de nuestras últimas obras publicadas, *Ukus*, con casi 95.000 palabras y apéndices (sin notas al pie o referencias bibliográficas), fue desarrollada en media hora. Ambos dimos el visto bueno a los ajustes en un día: ella no tuvo que revisar palabra por palabra y yo no me vi en la necesidad de preocuparme por tratar de homologar la estructura.

3. El *single source* y *online publishing* requiere de un esfuerzo corporativo para mayor accesibilidad

Una de las principales **debilidades actuales del *single source* y *online publishing*** —que no vale la pena negar— es que no cuenta con un entorno amigable. Incluso, el SSOP carece de *un* entorno, ya que por el momento se lleva a cabo a través de varios programas. Y estos, en su mayoría, carecen de un entorno gráfico o, en el mejor de los casos, de un entorno amigable para el usuario general. (Nótese cómo las necesidades editoriales se piensan como necesidades *generales*).

La mayoría de los usuarios estamos habituados a que con un solo *software* o con una paquetería (varios programas relacionados unos con otros) se delimite nuestro flujo de trabajo. Partiendo de esto, **el *single source* y *online publishing* requeriría de algún *software* o paquetería que ayude al editor a cernirse a esta metodología**.

El requerimiento no es disparatado, desde hace mucho tiempo he comentado la linda idea de un **Entorno de Edición Integrada (EEI)**. Este concepto no es original. En el desarrollo de *software* existen los Entornos de Desarrollo Integrado (IDE, por sus siglas en

inglés) los cuales son un conjunto de herramientas bajo un mismo ambiente de trabajo que pretende abarcar todas las necesidades que tiene un programador para llevar a cabo su trabajo. No es, para nada, un entorno «amigable» para el usuario promedio, pero sí es el que un desarrollador necesita.

Entonces, si un Entorno de Edición Integrada es posible (¿EPI, Entorno de Publicación Integrada?), este debería satisfacer la mayoría de las necesidades de los procesos editoriales. Es decir, sería un entorno no solo para el editor, sino también para el diseñador, el tipógrafo y el desarrollador.

Para nada sería, de nueva cuenta, un ambiente «accesible» para el usuario general, pero sí el que la edición necesita. Al final, el *software* utilizado actualmente para la edición también carece de esa «amigabilidad», hasta el punto donde actualmente hay editores que se niegan a aprender a usar InDesign, por poner un ejemplo.

Una idea semejante ha estado trabajando Adobe a través del dichoso InDesign. No es ningún secreto que este fue pensado para diseñadores editoriales de libros impresos, pero que ahora también el editor lo utiliza y, bajo las necesidades actuales, asimismo pretende ser una plataforma para la publicación digital, ya que permite la exportación a EPUB o crear aplicaciones.

Como muchos hemos experimentado, cuando se trata del trabajo del editor o de la creación de otros formatos, InDesign se queda corto. Este *software per se* no contempla necesidades de edición como son el cuidado en la uniformidad o el aspecto ortotipográfico.

Además, sin un conocimiento previo de tecnologías *web*, se obtienen EPUB sin la calidad necesaria (véase De XML a EPUB con InDesign). Ya no digamos de las aplicaciones: sus limitaciones son por demás evidentes en comparación a un desarrollo con

tecnologías nativas, híbridas o a través de motores de videojuegos.

Por este gran esfuerzo —necesario para cuajar todo un nuevo método en un solo entorno— varias personas opinan que **una inversión corporativa es necesaria**. Por ejemplo, Adobe desarrollando un nuevo *software* para suplir a InDesign en este nuevo panorama (podría decir que no lo dudo, Adobe también es conocido por descontinuar sus productos en aras de continuar en la «vanguardia»). Y solo a través de esta inversión sería posible tener, en relativamente poco tiempo, un «producto» —que ya no «método»— para cumplir satisfactoriamente con las necesidades de la producción editorial.

Sobre el futuro no puedo decir nada. Pero viendo el panorama, es cierto que el desarrollo no corporativo toma más tiempo. Por lo general, es trabajo realizado en horas libres; es menos conocido, porque carece de una infraestructura con poder mediático.

Asimismo, se enfoca en la creación de una diversidad de herramientas en lugar de un entorno que lo englobe todo, debido a que ese «todo» nunca es claro ya que cada usuario utiliza las herramientas según más le convenga y no bajo una idea de aceptación general sobre *cómo* se tiene que trabajar.

Sin embargo, también es cierto que el **trabajo de las comunidades abiertas de desarrollo** tiene más vida útil. TeX tiene más de treinta años y el HTML más de veinte. ¿Cuánto duró sino la hegemonía de QuarkXPress? Más flexibilidad de personalización y, claro está, la característica de prescindir de un pago obligatorio para su uso.

Llámenme iluso o soñador, pero si de accesibilidad hablamos, me cuesta mucho trabajo comprender su posibilidad cuando la neutralidad es casi inexistente, cuando es una estructura corporativa la que toma las decisiones.

¿Qué pasará cuando Adobe anuncie la descontinuación de InDesign? ¿Desaprender para aprender a utilizar otro *software*? La descontinuación no es infrecuente en el desarrollo de *software*, pero en comunidades abiertas y estables de desarrollo no tienden a cambios abruptos, sino una lenta y continua evolución. Quien aprendió TeX desde los ochenta poco o nada ha tenido que «actualizarse». Quien empezó a usar HTML en los noventa, las nuevas versiones le han dado más posibilidades de desarrollo.

Sobre esta entrada

- Título original: «¿*Single source* y *online publishing*, fuera de la realidad?».
- Título en el *blog* de Mariana: «¿Single source y online publishing, fuera de la realidad?».
- Fecha de publicación: 19 de diciembre del 2016.
- Redacción original.
- Revisión de Mariana.

Completud de una obra: ¿se acaba de editar con la publicación de un libro?

Entre la edición y la versión

El tiempo puede consumir un proyecto editorial. La obra se publica y quizá se encuentre alguna errata o tal vez no encontramos un error, pero, sin importar el caso, queda la sensación de que pudo haber estado mejor. Si hubiéramos tenido más tiempo, si no nos hubiéramos enfocado tanto en alguna cuestión, si…

La obra sale, se distribuye, se comercializa y la mayoría de los lectores ven al libro como un producto acabado que va a su estantería, que les agradará, o se arrepentirán de comprarlo y lo devolverán.

Pocos son los que saben que en la página cuarenta y cinco hay un callejón que solo se vio después de la publicación. O, más irritante, siempre quedará ese sabor de que la obra salió demasiado rápido, que quizá una lectura adicional no hubiera estado mal.

A muchas de las editoriales que hemos apoyado con el desarrollo de EPUB o con capacitación sienten estos síntomas. Algunos editores son francos y aceptan que la cantidad de tiempo no es suficiente para la cantidad de obras que *tienen que* publicar. Otros son más recelosos de sus procesos, pero la estructura de sus archivos los delata.

El formato, aunque por lo general «invisible», **demuestra el modo de trabajo e incluso anticipa dónde habrá inconsistencias antes de leer una obra**.

Pese a toda esa batalla librada tras bambalinas, sale *x* edición de una obra. Ya será para la siguiente edición donde se tomará más cuidado… Si es que con los años ese `archivo-final-final-final` no se pierde, aún se puede abrir o no se confunde con el `archivo-final3`.

¿Cuánto de este problema se debe a tratar de imponer un ritmo de trabajo a una profesión que ha sobresalido por la paciencia que se requiere para ejecutar un proyecto?

¿No será más bien alguna deficiencia metodológica? ¿Y si es falta de capacitación? Como sea, quienes se involucran en la producción de un libro en más de una ocasión sus herramientas representan un obstáculo.

Las versiones de una obra y el desarrollo de un proyecto editorial

Bajo este contexto, la tradición editorial se ve ante un nuevo fenómeno: las versiones. En los libros, el número de edición sirve para explicitar la completud de toda una serie de procesos.

Aunque sea parte del imaginario del lector o un ideal de quienes editan libros, **la completud rara vez significa que una obra ya no requiere más trabajo, más bien indica la resignación o imposición de tiempos a un proyecto**.

¿Cuántas veces en una presentación de un libro se indica ese deseo de haber tenido más tiempo? ¿Cuántas veces se admite que tuvo que cerrarse la edición, como si se tratase de una publicación periódica?

Sin embargo, en las versiones no sucede eso. En el ámbito del *software*, **el versionado solo indica el estado de desarrollo de un proyecto**. Cada nueva versión, sea para grandes o pequeños

cambios, o para corrección *o adición* de errores, no indica un estado de completud, sino el grado de madurez de un programa.

El *software* se percibe como un producto en constante crecimiento y mantenimiento, mientras que en la edición la obra se busca hacerla pública cuando se considera que está acabada.

Si en el número de edición su «completud» se vuelve ambigua —¿qué se «completó»: los procesos para publicar una obra o la cantidad de trabajo que esta requiere para ya no necesitar más cambios?—, las versiones comparten el supuesto de que un producto yace sobre un proceso evolutivo indefinido.

Al parecer una nueva versión siempre es mejor que la versión anterior. Pero las versiones recientes también pueden acarrear inconvenientes: la creación de *bugs*, un mayor consumo de recursos o la pérdida de soporte a dispositivos antiguos.

¿Qué tan «buena» es una nueva versión de un programa cuando este empieza a correr lento en nuestras computadoras?

Versiones y ediciones de una obra

Por fortuna, en el ámbito editorial nada se volverá más lento si además de ediciones se empiezan a trabajar con versiones. El uso de versiones en la edición permite la adición de mejoras o la corrección de errores sin tener que volver a iniciar todo un ciclo de trabajo.

Las versiones pueden permitir una mayor versatilidad en pos de la experiencia de lectura. La idea suena muy bien, pero su implementación puede ser un martirio técnico, de cuidado editorial o de índole legal.

Si continuamos en el contexto donde muchas de las editoriales batallan con sus propias herramientas y formatos para producir al

menos un producto, la complejidad técnica aumenta cuando se empiezan a meter más productos —claro está, si la metodología para publicar es la edición cíclica—.

El desafío aumenta cuando ya no solo se tiene que crear una edición, sino que de manera constante se han de actualizar los archivos según cada nueva versión. En el caso del impreso, si no se trata de impresión digital, de pocos y constantes tirajes, la idea de publicar cada nueva versión es simplemente una locura.

El caso de la publicación digital la situación no es menos compleja: entre las deficiencias metodológicas, las prisas y la falta de capacitación o de organización en los archivos, la actualización del archivo puede demorar varias horas o días; más si la persona responsable está poco familiarizada con las distintas plataformas. Esto obviando el visto de bueno de los responsables de la edición o, si se trata de instituciones, los trámites administrativos requeridos para la publicación de una obra.

Cambios y correcciones que introducen y producen errores

En el ámbito editorial, así como en una nueva versión de *software*, se pueden colar *bugs* que pueden provocar nuevas erratas o errores tipográficos en una versión más reciente de una obra. ¿Qué tal si quien introdujo el cambio presionó, por descuido, una tecla y coló una letra en el texto?

¿Qué tal si en una corrección, sin advertirse, se movió un párrafo, una página o una sección entera, creando callejones, viudas o huérfanas? Para los editores más exigentes, estos cambios requerirían, al menos, una lectura rápida de toda la obra… ¡pero no hay tiempo suficiente!

En el contexto multiformato puede también empezar a crearse

una disparidad en los diversos formatos. **El cuidado editorial también vela por la uniformidad**, y esta constante introducción de cambios poco a poco puede provocar que los formatos sean distintos.

Lo más probable es que la mayoría de los lectores no lo noten, pero en muchos casos no se trata del lector, sino de la mera insatisfacción del editor de saber que algo no cuaja, que poco a poco se pierde el control sobre la edición.

El ámbito legal no queda fuera. En una obra, por lo general, están involucrados una serie de contratos con autores y colaboradores, convenios con instituciones o distribuidores y trámites administrativos. Quizá todo este papeleo se pueda simplificar para que el uso de versiones no requiera mucha administración, pero si la «modernización» de contratos y convenios ya es un reto en manos del jurídico —por ejemplo, el ofrecimiento de licencias de uso—, hay una gran incógnita aún sin resolver: los ISBN.

El ISBN

El ISBN es una especie de cédula de identidad para cada edición y *cada formato*. Esta identidad se pierde con las versiones, porque entre versión y versión la obra ya no es la misma. Es decir, **en teoría, cada nueva versión requeriría un nuevo ISBN para ¡cada formato!** O abandonamos el ISBN y buscamos una solución estandarizada más dinámica: guiño a las llaves públicas o a las cadenas de bloques.

Los retos técnicos y legales que pueden representar el versionado es una cuestión que en el desarrollo de *software* se ha estado trabajando desde hace mucho tiempo. Por un lado, están las

licencias de uso de *software* que simplifican las cuestiones legales. Por el otro, en el desarrollo de *software* las versiones no son un reto técnico porque:

- solo se produce un ejecutable (para Windows, Mac o Linux);
- o se producen diferentes ejecutables a partir del mismo código fuente;
- o se tienen equipos independientes según la plataforma (un equipo para Android, otro para iOS).

En la edición, la producción de un solo formato de por sí ya es un reto debido a la falta de capacitación o de organización de los archivos. Las concepciones metodológicas multiformato, como la edición cíclica, son prácticamente desconocidas o ignoradas, por lo que no es posible crear diferentes formatos a partir de los mismos archivos madre. Solo queda la publicación de distintos formatos de manera independiente que, en muchos casos, es un despilfarro de recursos.

En el ámbito de *software* solo grandes empresas u organizaciones, o grupos de trabajo muy comprometidos, tienen los recursos suficientes para poder tener equipos independientes según la plataforma. Por ejemplo, Snapchat o Telegram tienen personal destinado para plataformas específicas; o bien, la comunidad de Linux que se encarga de adecuar el kernel según el tipo de arquitectura.

Si hacemos la analogía, en el ámbito editorial los únicos con la capacidad de actualizar sus formatos con pocos inconvenientes son las grandes casas editoriales. Por ejemplo, a Penguin Random House o a Grupo Planeta el uso de versiones más que un reto, sería un *plus* a sus ediciones, porque tienen la capacidad de destinar

distintos equipos. Pero a la mayoría de las editoriales —medianas, pequeñas o independientes— y, sin duda, a los autores que se autopublican este modo de producción no es el más conveniente.

El uso de versiones entre pequeños editores solo será factible cuando exista un cambio metodológico de fondo. Pero cuando eso sea posible, quizá el uso de versiones ya no sea necesario porque existe un modelo de desarrollo que podría ser más apto: el *rolling release*, que bien podemos llamar «edición continua».

Sobre esta entrada

- Título original: «¿Completud de una obra?».
- Título en el *blog* de Mariana: «Completud de una obra: ¿se acaba de editar con la publicación de un libro?».
- Fecha de publicación: 20 de febrero del 2018.
- Redacción original.
- Revisión de Mariana.

En varias entradas pasadas me he dedicado a diferenciar los distintos tipos de edición, como son la tradicional, la cíclica y la ramificada. (Para más información, consulta la tríada de artículos «Edición cíclica y edición ramificada» empezando por aquí.

De manera concisa, la **edición tradicional** es el conjunto de conocimientos, técnicas y métodos que se han empleado desde la invención de la imprenta y por medio de medios análogos.

La **edición cíclica** es una extensión de esta tradición, pero utilizando tecnologías digitales, por el cual el quehacer se centra en un dispositivo: la computadora; así como surge el multiformato: el *ebook junto* al impreso. **La edición es ya edición digital** en este contexto.

Por último, la **edición ramificada** es la propuesta metodológica que he estado trabajando y cuyo origen es el *single source and online publishing*, el cual pretende crear una diversidad de formatos a partir de un conjunto de «archivos madre».

La gran diferencia entre este último tipo de edición con el resto es que la producción de los formatos se da de forma paralela y descentralizada: **la edición ramificada invierte el problema de «a más formatos, mayor tiempo, mayor uso de recursos y menor control»**.

Pero no solo eso, **la metodología ramificada también pone sobre la mesa dos temas: la completud de una obra y la dependencia tecnológica implicada en la producción de publicaciones**.

Es aquí donde al fin me dedico a hablar sobre un término que he mencionado con brevedad en dos entradas anteriores: «edición continua».

Más allá de las ediciones de una obra

Como lo mencioné en otra entrada, **la completud de una obra, por lo general, es resignación**: los tiempos acotan las horas de trabajo, los recursos consumen su posibilidad de realización y la carencia de conocimientos técnicos limitan sus características. La producción de una publicación es un proceso arduo y, en más de una occasion, no se desarrolla según lo planificado.

Esto da pie a diversas implicaciones:

- En un **contexto multiformato** abre la puerta para un mayor descontrol y pérdida de calidad, debido a que el personal tiene que estar capacitado sobre las características de cada formato.
- En un **contexto cíclico** se traduce en tener que emplear menos recursos y tiempo para producir más formatos en un mismo periodo.
- En un **contexto propietario** quiere decir que se ha de invertir en *software* y *hardware* cada vez más especializado para dar un cumplimiento afín a la calidad y el calendario estimados.

Debido a estas cuestiones **es común que las personas dedicadas a la edición, por lo general, recuerden la publicación de una obra como un proceso traumático o al menos insatisfactorio**: siempre se pudo haber hecho más, pero los tiempos, el presupuesto, la

computadora, el personal…

La mayoría de los editores ya tienen serios problemas al momento de lidiar con las diversas ediciones de una obra y esta dificultad se exacerba cuando en el ámbito digital existe la posibilidad del uso de versiones. Si cada edición indica la fecha de completud de una obra, en el versionado se hace patente el nivel de maduración de un proyecto.

La diferencia no es nimia. **En el número de edición se apela a la completud de la obra, mientras que en el número de versión se explicita su incompletud**. En el versionado la completud es una idea que busca su cumplimiento mediante el crecimiento paulatino de un proyecto.

La completud deja de ser concreta y pasa a ser una guía para el rumbo que tome una publicación. La completud exhibe que la «obra» no es el puerto destino, sino que esta está en constante «evolución», cuya prioridad es la gestación del «proyecto». En fin, la completud es el ensueño de la imaginación del editor.

El versionado da dinamismo, pero también acarrea un mayor nivel de compromiso. Siempre es una ventaja suponer que la edición marca el fin de un proyecto, aunque sea solo temporal: es la calma después de la tempestad.

No obstante, en el versionado el proyecto permanece abierto: a cada instante se está a la intemperie, por lo que un constante mantenimiento es necesario para evitar el naufragio. **Cuando un proyecto deja de tener versiones, en lugar de decir que se ha completo, se indica que se ha abandonado**. Es decir, en el versionado la completud nunca llega.

Este dinamismo puede ser menos accidentado si se tiene una metodología que permita automatizar y hacer menos tortuoso cada uno de los procedimientos implicados en la publicación de una

obra. **La edición ramificada es una propuesta para subsanar las dificultades en esta dinámica**, aunque implica el aprendizaje explícito de un método de producción y las técnicas y los conocimientos que vienen embebidos en ello.

Más allá de las versiones de una obra

Pero **cuando esta conciencia metodológica y el control técnico son alcanzados, el versionado deja de ser necesario** —y no solo hablo sobre el método ramificado, sino cualquier tipo de metodología que permita los mismos resultados—.

No es que el versionado se torne anticuado o inconveniente, sino que en este contexto es posible alcanzar tal grado de dinamismo —un movimiento vertiginoso de constantes cambios sobre el proyecto— que el versionado aumenta rápidamente. Por lo general, la idea de manejo de versiones busca ir paso a paso, en cada nueva versión se pretende incluir una serie de mejoras o arreglos en un solo paquete que vuelve a instalarse nuevamente.

Sin embargo, cuando los cambios son tan constantes, puede incluso resultar confuso qué mejoras o arreglos incluir en la siguiente versión, así que por pragmatismo o pereza se recurre a la liberación continua. En lugar de crear paquetes nuevos —todo el proyecto una y otra vez—, se actualizan los paquetes —el proyecto es el mismo y solo se descargan sus cambios—.

«Libera pronto, libera continuamente» es una frase atribuida a Eric S. Raymond, personaje clave para el movimiento del código abierto. Y lo que implica es la publicación constante de cambios que aunque parezca caótico, marca un ritmo de desarrollo constante que permite a los demás tener un contacto de primera mano con la gestación de un programa.

En inglés, este modelo de desarrollo se conoce como «*rolling release*». Y la idea de «rodante» hace más clara la imagen de la vida de un proyecto como una bola de nieve: movimiento cada vez más robusto y acelerado, que continúa hasta colapsar.

El *rolling release* en la edición

La publicación continua puede tener un mal sabor de boca para muchos editores. Implica que cada vez que se hace un cambio, la obra de nuevo se publica: una práctica que atenta directamente a la idea de ediciones, pero también a la de versiones.

Pero ¿cuántos no han hecho «trampa» y han colado algunas correcciones en el archivo y aun así la siguen catalogando como la misma edición o versión? **El dinamismo en la edición no es una novedad**, pero lo que se admite en un modelo de liberación continua es que la obra ya no es la misma. Caso contrario es fingir la identidad de la obra al conservar el número de edición o de versión, cuando la publicación ya se ha modificado.

La liberación continua es franca con el lector: «la obra no está completa, sigue en constante cambio, disculpa las molestias». Incluso en su colapso: «esta obra ya ha ido demasiado lejos que no puedo mantenerla más». El abandono ya no sabe a fracaso, sino a un desvío de intereses o de energía. Además, si el proyecto es abierto implica una apertura para que otras personas le den continuidad.

Ejemplos de esta liberación continua la podemos ver en el historial de cambios de Pecas, *una* propuesta concreta de la metodología ramificada de publicación y que a esta fecha tiene ya 573 actualizaciones en tan solo dos años.

Otro ejemplo de edición continua es *Edición digital como*

metodología para una edición global, la antología de estas entradas que a la fecha cuenta con 124 actualizaciones en casi dos años y 59 actualizaciones desde su primera publicación, hace cuatro meses y medio.

¿Imaginan lo inverosímil que sería indicar que en una obra que aún no ha alcanzado ni medio año de vida ya cuenta con 59 ediciones o versiones?

Identificación en la avalancha

Como se puede observar en los historiales de cambios, **la liberación continua no es sinónimo de pérdida de control**. Al contrario, cada cambio tiene un identificador único universal (UUID), fecha —incluyendo huso horario—, usuario que realizó el cambio —con correo electrónico incluido— y hasta qué modificación concreta se hizo a cada archivo.

Este gran dominio es gracias a git, un tipo de control de versiones cuyas características generales se han catalogado como el octavo elemento metodológico de la edición ramificada.

Siempre es posible saber el cuándo, el qué y el quién. El nivel de detalle es tal que incluso da más información sobre la «evolución» de una obra que la consulta de formatos finales, también conocidos como ediciones o versiones; o al menos facilitan su consulta.

Además, este proceso de identificación visibiliza dos cuestiones. La primera es que no hay método más sencillo para identificar y mencionar la antigüedad de una obra que su fecha. El uso de UUID es mucho más exacto, ya que en un mismo día puede haber varias actualizaciones, pero es poco legible por humanos, *¿qué significado tendrá un conjunto de caracteres alfanuméricos?*

Esta falta de legibilidad también es compartida con la numeración para las ediciones o versiones: «primera edición», «versión 2». ¿Qué quiere decir eso para conocer la antigüedad de una obra? ¡Qué año data esa dichosa «primera edición» o «versión 2»!

La obra evoluciona cuando deja de ser el centro y la publicación se desecha…

Esto abre paso a la segunda cuestión. Como también puede observarse en el historial de cambios de la antología, muchas de las modificaciones no se hicieron directamente a los formatos publicados: unas son adiciones de nuevas entradas, otras son actualizaciones de los archivos madres y algunas más son modificaciones del *script* que produce los formatos o implementaciones de nuevas funcionalidades de Pecas.

El control en la liberación continua no solo es sobre la obra, sino del proyecto, sobre el material implicado para producirla. El bibliófilo no solo puede ver de manera más amena la evolución de la obra, sino que también tiene acceso a los procesos que subyacen detrás de su producción.

La cuestión no es paupérrima. **La tradición editorial ha hecho del libro su centro de atención** y agregaría que, más bien, lo ha hecho **a un formato en particular: el impreso**. La imagen no parece problemática e incluso su crítica parece contraintuitiva: si tu gremio se dedica a publicar obras, ¿qué problema hay con que el libro sea el centro de ese universo?

El «librocentrismo» en la industria editorial

El empeño por querer tener formatos finales —los únicos elegibles para la publicación— hace que la persona editora se centre más en el producto final —en muchos casos como «mercancía»— que en los pasos necesarios para obtenerlo. O bien, cuando nace la preocupación por los procesos necesarios, **el enfoque centralizado en la obra supone que a cada formato le corresponde su propia vía de gestación o, peor aún, que un formato *precede* a los demás**.

El «librocentrismo» hace de la publicación el desenlace *necesario* de los procesos editoriales. Para un público general, esto se oculta hasta el punto de minimizar el quehacer editorial: «¿por qué tarda tanto si solo es un montón de papel con tinta o lenguaje de marcado?».

Ante las personas que dictaminan el curso de una editorial o incluso de toda la industria en una región, es carencia de capacitación y falta de recursos: «se busca persona con conocimientos de *x* programa privativo; del presupuesto cotiza un taller para usar *y* paquetería de *software* que en *z* feria anunciaron como la panacea editorial; pide apoyos gubernamentales o internacionales porque cada tres años *tenemos* que actualizar nuestro equipo de cómputo».

En el librocentrismo la dependencia tecnológica no importa, siempre y cuando ese *software* o *hardware* cumplan con el cometido de publicar la obra. Una paquetería de diseño o de oficina se vuelve omnipresente, aunque en realidad pocos la dominan. Un conjunto de programas es confundido con un método *único* de producción de libros.

Y si un proyecto falla, el programa, la máquina o el personal es quien tiene la culpa; cuando el problema es que la falta de capacitación nunca ha sido en relación con los formatos deseados,

sino la evasión de ver que **la edición no solo es una tradición o una profesión, sino también un método**.

Si preparas a una generación de editores o diseñadores para utilizar programas propietarios determinados es casi seguro que una vez terminada su formación esos conocimientos sean obsoletos —ignoremos un poco la diferencia entre el uso de estos programas en la escuela y en el trabajo—.

De nueva cuenta hay que buscar capacitaciones, con la cruda realidad donde lo que no ha cambiado es la dependencia a que alguna compañía o institución te proveerá del *software* y *hardware* necesario para hacer tu trabajo.

Pero si nos centramos en el método, en cómo publicar una obra en diversos formatos con la mayor automatización y calidad posible, se hará patente que lo relevante yace en los archivos de origen y en el camino que va de ellos al multiformato.

Un formato determinado deja de tener mayor jerarquía. Los formatos dejan de «competir» entre ellos. Los procesos de producción multiformato ya no se perciben como ajenos. **La publicación deja de ser el objetivo: es desechable**.

El centro se vuelca al cuidado de los archivos madres y al mantenimiento de los procesos automatizados de publicación multiformato. Los formatos finales, esos EPUB, MOBI o PDF a publicar, se producen incesantemente y se eliminan sin el temor de haber perdido algo. No hay merma porque el valor de la obra reside en sus orígenes, la publicación es solo una muestra de los esfuerzos aglutinados en un objeto, que puede reproducirse sin problemas.

Como editores, ¿cuántos proyectos yacen ahí en nuestras computadoras, discos duros externos, nubes o repositorios que han quedado abandonados? En la pretensión de publicar un texto hemos minado sus posibilidades al atropellar, una y otra vez, la

metodología necesaria para su gestación.

Enseña a una generación de diseñadores y editores a concebir la edición también como un método y no habrá *software* o *hardware* que los limite. Mejor aún, por esa libertad dejarán de estar a la espera de compañías o instituciones porque cada quien, como persona, equipo de trabajo o editorial, será el arquitecto de su propio método.

Un giro en la edición es necesario. Al tener conciencia clara del método, la obra evolucionará de manera favorable y la publicación tendrá los estándares de calidad deseados. **En el anhelo por cosechar sus frutos, la tradición editorial ha hecho del libro un fetiche**. La edición continua y más específicamente la edición ramificada y libre podrían ser *uno de esos* giros.

Sobre esta entrada

- Título original: «Edición como edición continua».
- Título en el *blog* de Mariana: «La edición como edición continua».
- Fecha de publicación: 2 de julio del 2018.
- Redacción original.
- Revisión de Mariana.

Cómo las herramientas analíticas pueden mejorar el quehacer editorial

Unas de las características que distinguen al cuidado editorial son la meticulosidad puesta en la redacción, la ortotipografía, la verificación de datos y la uniformidad en la estructura y los estilos.

Sin embargo, muchas veces **mantener la calidad se vuelve una tarea de difícil cumplimiento**: tiempos acotados, bajo presupuesto, proyectos que han pasado por muchas manos o una mera carencia de organización o de las habilidades necesarias son algunos elementos que van en detrimento de la calidad editorial.

Las diversas técnicas tradicionales o las recomendaciones de editores con un amplio legado son fundamentales para subsanar estas debilidades. Pero no tenemos que detenernos ahí, ahora que **la edición es edición digital**, existe una diversidad de programas que pueden ayudarnos a tener un mayor control en el quehacer editorial.

El antiguo aliado y enemigo: los diccionarios

Con el surgimiento de los procesadores de texto nació una característica que modificaría la manera en cómo un escritor revisaba sus propios textos. Me refiero a los correctores ortográficos que en el documento subrayan con rojo alguna palabra mal escrita. O eso creíamos las primeras veces que usábamos la herramienta.

¿Cuántas veces no se dio clic en «Corregir todo» o «Reemplazar

todo» y se hacían los cambios en lo que el corrector consideraba erróneo pero, en realidad, simplemente eran palabras que no estaban en su diccionario?

Vaya dolor de cabeza darnos cuenta que nombres, regionalismos o ciertas conjugaciones desaparecían para abrir paso a un documento casi ininteligible…

El entusiasmo de la mayoría de quienes usaban los correctores —«para qué queremos editores, si el Word y Amazon ya hace eso», dice alguien cuando habla con un protoeditor cualquiera— fue opacado con el recibimiento negativo dentro del mundo editorial. ¿Qué clase de editor usa semejante herramienta —aunque muchas veces se trata de un gusto culposo—?

Hay que distinguir entre los propósitos del corrector ortográfico de su base y sus posibles usos; a saber, los diccionarios. La manera en cómo el corrector identifica las palabras, para posteriormente indicar si es «errónea» y de ahí permitir la sustitución, es gracias a una colección de palabras, también llamadas diccionarios. Un ejemplo de las primeras diez entradas de un diccionario en español de México es el siguiente:

```
57160
=======
a
ababa/S
ababol/S
abacería/S
abacero/GS
ábaco/S
```

```
abada/S
abadejo/S
abadengo/GS
abadengo/S
```

Este diccionario tiene poco más de cincuenta y siete mil entradas, y cualquier persona podría seguir llenándolo o crearlo desde cero. Pero para fines editoriales esto no es común, ni tampoco necesario. Lo importante es que ya se tiene una lista de palabras existentes en un idioma, *sin importar que no esté completa*.

Una ayuda para el trabajo monótono

¿Para qué nos puede servir este diccionario, si lo usamos de manera independiente a los correctores? Muy sencillo: **automatización de procesos monótonos**. Hay que tener cuidado, como es común cuando se habla de «automatización» se tiende a entender que alguna máquina hará todo el trabajo por nosotros, pero este no es el único significado.

La automatización como sustitución del trabajo humano es, por así decirlo, la acepción fuerte del término. El propósito de los correctores es automatizar en este sentido el trabajo de un corrector humano —nótese que no de un editor humano—.

Sin embargo, **la automatización también puede ser un auxiliar del trabajo humano**, principalmente en tareas monótonas, **una acepción que podemos catalogarla de «débil»**.

Los «diccionarios» que usan estos correctores no pretenden

reemplazar el trabajo humano, incluso tampoco pretenden sustituir a lo que durante siglos hemos entendido por «diccionario» —¡ni siquiera tienen definiciones!—. Su propósito es ser un auxiliar para distintas actividades, principalmente de análisis, pero también para mejorar la calidad editorial.

Aplicación práctica en el proceso de edición

Su uso dentro de la edición es sencillo. Como es común, o al menos deseable, en el quehacer editorial, varias personas revisan un texto para corregir erratas y otras cuestiones. Debido a que uno es ciego ante sus errores, la lectura de diferentes personas permite encontrar deficiencias pasadas por alto. La norma es: *muchos ojos, pero solo dos manos*. Es decir, el editor a cargo cotejará y aplicará los cambios que diversos editores o correctores anotaron.

No obstante, también es común que en el camino puedan añadirse dedazos u otros horrores ortográficos, cuya relectura puede resultar costosa o consumir mucho tiempo. Para estos casos finales, quizá el uso de un diccionario puede ayudar a mostrar posibles erratas en un texto mediante los siguientes pasos:

1. La persona a cargo de la edición manifiesta que el trabajo ya está listo para su maquetación o desarrollo de EPUB.
2. Antes de pasarlo al departamento de diseño o de desarrollo se utiliza un programa para generar solo una lista de palabras únicas que no se encuentran en el diccionario. Ejemplo de programas tenemos a hunspell o a aspell (ambos gratuitos y libres).
3. Se revisa esta lista para descartar las palabras que sí son correctas (incluso pueden añadirse al diccionario para

evitar los falsos positivos).[1]

4. Las palabras restantes se cotejan en su contexto y, de ser erróneas, se corrigen.
5. Terminada esta revisión, se sigue con el flujo normal dentro de los procesos editoriales.

Aunque parece que esto consume mucho tiempo, en realidad el cotejo es muy rápido, ya que solo se trata de una lista donde rápidamente es posible eliminar los falsos positivos y así concentrarnos únicamente en los casos dudosos. Esta forma de cotejo es mucho más rápida que estar saltando sobre el texto, más aún si paulatinamente se van agregando otros términos.

De esta manera tenemos al menos mayor certeza sobre la calidad ortográfica de nuestro proyecto, pero ¿para qué quedarnos ahí? Este modelo de trabajo puede servir para otras características más allá de la ortografía.

Lo que al autor no le importa pero que da dolores de cabeza: los enlaces

Por experiencia, he notado que **la mayoría de los autores tienen un completo desinterés por el estado actual de los enlaces** que ponen en su documento, incluso entre aquellos que aman los enlaces a páginas *web* o referencias cruzadas.

Cuando se trata de enlaces internos, en más de una ocasión me topo con que el elemento referenciado ya no existe o que su identificador ha cambiado. En el caso de los enlaces externos, es casi seguro que al menos la mitad de los enlaces estarán mal escritos o serán ya inválidos.

Cuando la obra solo tiene un par de vinculación, no hay ningún

problema con revisarlas manualmente. Pero ¿qué pasa cuando estamos ante una obra muy extensa, con más de cincuenta enlaces y muchos de ellos repetidos a lo largo de diferentes secciones?

En el peor de los casos se ignora por completo y ya será el lector quien se dé cuenta de esta falta. En la mejor situación se revisa cada enlace y se corrige de manera manual, insumiendo desde una jornada hasta una semana en completarse el trabajo, lo que al final ya no es tan bueno, más si los enlaces externos cambian su estatus de manera constante…

Para estos casos, se pueden seguir los mismos pasos de los diccionarios, pero en lugar de analizar palabras, se analizan enlaces y, de manera automática, se verifica su estatus.

Esto también genera una lista que no solo permite saber si un enlace es válido o no, sino también los motivos por los que son inválidos, pudiendo descartar así si es un enlace muerto, el servidor actualmente está saturado o es un llano error de sintaxis.

Así ya tenemos un mayor cuidado en la ortografía y en los enlaces, pero ¿para qué detenernos ahí?

Las analíticas como parte del quehacer editorial

Por lo general, las analíticas se asocian al análisis de mercado. Por ejemplo: ¿qué dispositivos usan los lectores?, ¿cuáles son los formatos más comunes?, ¿qué tipo de obra se consume más?

Incluso a casos que ya rayan en la invasión de la privacidad, como horarios de lectura, velocidad de cambio de página, palabras más buscadas en el diccionario, palabras subrayadas y más hábitos de lectura —*¿o no, Amazon?*—.

Pero **las analíticas en el ámbito del libro también pueden ayudarnos a mejorar el cuidado editorial**. Como ejemplos

tenemos:

Palabras más comunes en la obra

Permiten dar etiquetas más atinadas e incluso descubrir elementos que en una lectura lineal no habíamos sospechado.

Por ejemplo, que en la antología de estas entradas el programa más mencionado es InDesign, pese a que es público y notorio mi preferencia a no usar paquetería de Adobe y, en general, cualquier *software* que no sea libre o abierto…

Palabras sin identificar

Es decir, palabras que el análisis no pudo reconocer como meras letras o cifras, sino una combinación de ambos e incluso con otros caracteres.

Esto permite evidenciar de manera sencilla posibles deficiencias en la redacción o arcaísmos.

Por ejemplo: «post-moderno» en lugar de «posmoderno» en un texto donde ambos términos se usan indistintamente. En este caso no solo corregimos arcaísmos, sino que obtenemos una mayor uniformidad.

Palabras con versal inicial

Esta clase de listado nos permite cotejar rápidamente posibles variantes en la escritura de un mismo personaje.

Los ejemplos clásicos son los nombres rusos. ¿Cuántas veces nos ha pasado que en una obra se mezclan indistintamente dos o más variantes, como «Trotski» y «Trotsky»?

O peor aún, ¿variantes incorrectas como «Nietzche» o «Nietsche» en lugar de «Nietzsche»?

Índice de diversidad

Esto indica la frecuencia con la que una nueva palabra aparece en la obra; entre más tienda a cero, más diversidad tiene la obra; entre más extensa, la tendencia se aleja del cero.

Más que mera curiosidad entre la comparación de estilos de diversos escritores, en casos extremos podríamos detectar potenciales plagios.

Por ejemplo, cuando la obra de un autor tiene un índice de diversidad muy distinto a lo que es habitual en sus escritos: o es algo muy experimental, o en poco tiempo olvidó o aprendió nuevas palabras, o bien otra persona escribió la obra.

Estructura de la obra

Es posible tener un listado de los elementos que componen nuestra obra, como párrafos, encabezados, bloques de cita, etcétera, sin importar que sea para un formato impreso o digital. Esto puede ayudar a detectar inconsistencias o errores en la estructura.

Por ejemplo, en el análisis de *Edición digital como metodología para una edición global* existe una itálica no semántica (etiqueta `_`) cuando yo siempre empleo itálicas semánticas (etiqueta `*`), resulta que en la legal por accidente usé una etiqueta en lugar de la otra.

Otro ejemplo se da cuando, mientras revisamos las entradas Mariana y yo, algunos encabezados cambian de jerarquía; sin este

tipo de revisión, las posibilidades de dar con esta inconsistencia —y corregirla— serían mucho menores.

Entre las herramientas de Pecas tenemos una para crear esta analítica que aunque aún le falta pulirse, ya ha demostrado su pertinencia al momento de mejorar la calidad de los libros con los que trabajamos.

Sobre esta entrada

- Título original: «Herramientas para el quehacer editorial».
- Título en el *blog* de Mariana: «Cómo las herramientas analíticas pueden mejorar el quehacer editorial».
- Fecha de publicación: 24 de mayo del 2018.
- Redacción original.
- Revisión de Mariana.

Dolores de cabeza frecuentes para un editor de *ebooks*

Al momento de desarrollar un libro electrónico hay varios elementos que pueden fallar. Lo primero que sale a flote es la confianza sobre la capacidad de quien hace el libro. Sin embargo, en varias ocasiones el *ebook* no se muestra como debería y esto se debe a cuestiones ajenas a la calidad técnica del editor.

Quien encarga la creación o conversión de un libro digital debe conocer las siguientes particularidades que presenta la edición digital.

En esta entrada se abordan cuestiones básicas que atañen a la producción los *ebooks*:

- La inexistencia de «página» en los libros digitales.
- La limitación de los diferentes dispositivos en cuestiones tipográficas.
- Las capacidades del formato EPUB.
- Qué hace un renderizador y cómo afecta a un *ebook*.
- Los distribuidores y su papel.
- Los formatos propietarios, el DRM y la minería de datos.

1. Una analogía (errónea): la página

En la edición de libros impresos la página es el paradigma para la lectura y disfrute de una obra. Tras bambalinas, la página también es el modelo para la corrección de errores.

¿Cuántas veces hemos leído «te mando los siguientes cambios:

en *la página 5…*»? La página es muy útil para moverse a través de la obra. Más cuando la obra no se lee, sino que se emplea como referencia.

Un tipo útil de referencia es la que indica donde hay erratas, errores de formación y más. Esto bajo el supuesto de que la página tiene una dimensión constante en todos los dispositivos de lectura.

Para desgracia del control editorial, en las publicaciones electrónicas la página nunca tiene el mismo tamaño. La página, como un lienzo o un contenedor, no es una analogía lo suficientemente fuerte para representar el rol que desempeña en la edición.

La página implica una idea de un orden perenne. Mientras que el libro sea libro, la página siempre permanecerá igual, se raye, se moje o se rompa.

En un libro electrónico este carácter fijo ya no existe. La «página» digital es solo una remembranza del papel que desempeñaba durante la lectura. La tradición editorial trasladó la publicación como pergamino a una conformada por varias páginas.

En la edición digital la «página» es solo una convención que da al lector el mismo tipo de dirección al que está acostumbrado: izquierda / derecha → arriba / abajo → anverso / reverso.

Lo más grave que este equívoco no es la dificultad añadida para poder localizar las referencias a errores. ¿Qué hay de las viudas y huérfanas? ¿De una imagen y su pie de foto anclado? ¿De los callejones y los desbordamientos?

La «página» del libro digital no ofrece soluciones tipográficas

En la publicación digital, al ser variable el tamaño de la «página», no se puede dar solución definitiva a estos problemas en la

formación. **Lo que funcione para un tamaño de pantalla o dispositivo no necesariamente servirá para otros tipos de dispositivos**. Incluso puede ir en detrimento de la experiencia de lectura, como son indeseables saltos de línea o desbordamientos.

La cuestión reside en que los recursos necesarios para solventarlos no los justifican: **el lector promedio es ciego a estos errores de formación**. Esto, asimismo, quiere decir que estamos ante dos tensiones en el constante devenir del desarrollo de *ereaders*.

Por un lado, lo que el programador y el editor consideran relevantes. En no pocas ocasiones quien programa tiene otras prioridades: concibe al libro de modo distinto a quien lo edita.

Por el otro, no puede dejarse de lado que el principal desarrollo de los *ereaders* se da en el mundo anglosajón. Las prioridades en el cuidado editorial en inglés difieren con las de nuestra lengua. Ejemplo: la levedad con la que se permiten las viudas y las huérfanas.

Cuando nos adentramos al **cuidado editorial de publicaciones electrónicas en español** no queda sino aceptarlo. La página y todo su cuidado en la formación en nuestra lengua queda muy mermada cuando hablamos de «páginas» en los *ereaders*.

¿Qué pasaría si en lugar de rendirnos decidimos, conscientemente, comenzar el desarrollo de estándares en *ereaders* amigables a la tradición editorial en español?

2. El culpable directo: el editor-diseñador-desarrollador

Una vez aceptada a regañadientes la limitación que nos da la «página» del *ebook*, acontecen otros problemas. En principio, estos son achacables a la persona que desarrolla la publicación.

La falta de profesionalización en la creación de publicaciones electrónicas provoca que muchas veces la calidad posible no sea alcanzada. Entre el cuidado del texto como contenido, luego como diseño y hojas CSS, hasta como estructura y etiquetas HTML acontece una constante transposición de dimensiones del mismo texto.

No es sencillo desarrollar el ojo para **tener capacidad de percibir al texto en estas tres dimensiones al unísono**. Lo que parece una tarea sencilla, se convierte en un infierno para quien tiene un primer acercamiento a la publicación digital.

Como lectores o personas ajenas a los procesos requeridos para hacer un *ebook* es sencillo juzgar un libro por lo que se ve. Como es común, un libro electrónico tiende a tener una formación más sencilla, así como es estéticamente menos placentero. De ahí se concluye que su creación ha de ser sencilla.

Nada más descontextualizado que esta suposición. Si el *ebook* tiende a ser «feo» o «simplón» no es porque el formato no permita más, sino debido a que su dominio exige un largo camino para quien aprendió a editar a través del sendero visual del diseño editorial.

El formato estándar del libro electrónico permite muchas posibilidades interesantes. Sin embargo, el desarrollador de *ebooks* no solo tiene que aprender esto, sino también tiene que repensar el diseño a través de hojas de estilo y mantener la calidad editorial en un contexto de «páginas fluidas».

A esto se suma que el trabajo hecho en una publicación electrónica, por lo general, es menospreciado por quienes se dedican a la publicación de impresos. Como resultado tenemos que muchas veces, aunque ya se cuente con la capacidad de mejorar el *ebook*, no se lleva a cabo debido a que el tiempo requerido y el poco reconocimiento no lo justifica.

¿Qué tal si ya, por fin, nos libramos del menosprecio hacia el *ebook* y, en su lugar, vemos el carácter tripartito de su gestación y la necesidad de hacerle frente?

3. El diligente (y menospreciado): el EPUB

Entre los formatos digitales, el EPUB es un estándar en la industria con mucha potencia. Su aspecto más conocido es la posibilidad de **embeber objetos multimedia**.

Otra característica relevante es la **capacidad de poder incluir JavaScript**. Con este lenguaje de programación se pueden añadir diversas funcionalidades a un libro. Por ejemplo: animaciones, formularios, modificación de estructuras y hasta videojuegos.

A esto sumamos el **soporte para personas con deficiencia visual**, a la creación de fórmulas matemáticas e incluso la experimentación para la adición amena de partituras.

Con las hojas aurales (hojas de estilo auditivas) es posible escuchar la obra de manera más natural. Las fórmulas matemáticas ya no tienen que ser imágenes, sino texto. Del mismo modo, las partituras pueden seleccionarse nota a nota y hasta escucharlas con reproductores MIDI.

Y no lo es todo, la estandarización del EPUB da pauta para que **todas estas posibilidades puedan verse igual y del mismo modo en cualquier *ereader***. Y aquí es donde nos aproximamos a terreno escabroso…

¿Qué pasa si antes de seguir nos detenemos un poco y nos preocupamos por tratar de contemplar todo lo que el EPUB tiene por ofrecer, en lugar de desaprobarlo y aclamar necesidad de nuevos formatos (propietarios)?

El EPUB tiene mucho por ofrecer, pero sus posibilidades por lo general se han mermado por los renderizadores. Todo dispositivo o programa que se utilice para leer un *ebook* es un renderizador.

De las instrucciones y funciones incluidas en un EPUB, el *ereader* o el *software* procede a pasar de ese código a un resultado legible. Esto es el proceso de renderización: la **generación de una imagen a partir de un conjunto de comandos**.

Por desgracia, aunque existen estándares sobre cómo debería renderizarse el *ebook*, por limitaciones de *hardware* o decisión del desarrollador, esto queda como sugerencia.

Su efecto ya es muy conocido para muchos de nosotros: **el libro nunca se ve de la misma manera**. Por ejemplo, hay ausencia de versalitas, falta de corte silábico o diferencia en el interlineado o en los márgenes.

Un dispositivo antiguo no podrá dar el mismo desempeño cuando muestra las características más complejas e incluso, simplemente, no las ejecutará.

También existen otros casos en que desarrolladores como Apple o Amazon restringen algunas posibilidades. Ejemplos: Amazon no permite la adición de ninguna funcionalidad de JavaScript y el soporte de CSS tiene sus limitantes; Apple bloquea el uso de formularios que ni con JavaScript pueden eludirse.

La aplicación de Google, entre las más populares, es la menos limitada. No obstante, de vez en cuando aparece un *glitch* en el que se crean saltos de línea forzados. El párrafo a la vista se ve mal formado aunque su estructura esté limpia.

Esta disparidad en el soporte en no pocas situaciones **provoca una tensión entre quien desarrolla el *ebook* y su cliente.** En más

de una ocasión, tanto el cliente como yo, hemos tenido que ser lo suficiente pacientes para demostrar **que la ausencia de una característica o la aparición de un error en la formación no es por la mala calidad del código sino una cuestión del renderizador**.

Por estos motivos, el renderizador menos problemático que he encontrado es Calibre. En particular, es el que siempre recomiendo porque además es *software* libre. La interfaz puede parecer burda, pero su respeto a los estándares es muy elevado.

¿Que tal si, de una buena vez, caemos en cuenta que las posibilidades del *ebook* están constreñidas a las características de cada uno de los dispositivos?

5. Segundos cómplices: los distribuidores

El *ebook* puede estar ya desarrollado para sacarle su mejor partida en una amplia gama de dispositivos. Sin embargo, aun así nos topamos con **dos grandes inconvenientes de los distribuidores**.

A. La conversión a formatos propietarios

El primero es que algunos de ellos —guiño de nuevo a Amazon y Apple— no ponen a la venta el formato estándar que hemos trabajado. En su lugar, **los convierten en sus formatos propietarios**.

La argumentación es que sus propios formatos están optimizados para sus renderizadores. Por ello, lo aconsejable es utilizar sus herramientas de creación de *ebook* para producir sus formatos de manera directa.

Además de que sus formatos propietarios son herederos

directos de estándares abiertos, estos incluyen características adicionales que habilitan el DRM y la minería de datos.

A esto se suma el aumento de horas de trabajo para poder crear una publicación en distintos formatos si se opta por la vía de la bifurcación del proyecto en lugar de la conversión.

La inutilidad del DRM

El candado digital que evita la reproducción no autorizada de una obra (el DRM) es, en mi opinión, una de las cuestiones en las que la edición digital ha despilfarrado una gran cantidad de recursos. **Este candado nunca se ha mostrado lo suficientemente seguro** para las personas que saben cómo eliminarlo.

El aprendizaje sobre cómo hacerlo no es complejo. Con un uso sencillo de la terminal y un par de *plugins* de Calibre es posible «liberar» las obras.

En lo personal, es un procedimiento común cuando adquiero algún libro en Amazon o en sitios que cuentan con DRM de Adobe. No por la supuesta «piratería» que el DRM pretende evitar, sino porque me desagrada la idea de que su uso es limitado a pesar de haber comprado el *ebook*. Este desagrado muta en necesidad cuando estas tecnologías propietarias no dan soporte a sistemas operativos abiertos.

Minería de datos o la recolección de información

La minería de datos es una cuestión muy delicada y que casi no se ha comentado en discusiones relativas a la edición y la publicación digitales.

Bajo el eufemismo «análisis de hábitos de lectura» lo que varios

renderizadores llevan a cabo es una recolección de información que posteriormente será procesada por sus compañías desarrolladoras.

Con todas las relaciones encontradas es posible **monetizar la manera en cómo leemos libros**. ¿Maravillado porque Google, Amazon o Apple siempre te sugiere buenos libros? ¿Asombrado de las coincidencias entre lo que lees y la publicidad que te muestra? No es fortuito, **ellos buscan el medio para venderte algo, incluso a través de lo que lees**.

B. La aprobación de la publicación de ebooks

El segundo inconveniente no está en el lado del usuario hecho lector, sino en el del editor. Para quienes suben libros a distintas plataformas, en más de una ocasión, es un martirio que la publicación sea aprobada.

Dos ejemplos. Uno, Apple solo permite la subida de libros a través de iTunes Producer que, como pueden adivinar, solo está disponible para macOS. Y Amazon ofrece mucho menos regalías cuando el *ebook* no es exclusivo de su plataforma.

Las soluciones me parecen limitadas. Una consiste en pedir prestada una Mac para subir el libro. La otra es que, como Amazon es dueño de su plataforma, uno elige si acepta o no sus condiciones.

Como usuario y como editor **no deberíamos estar limitados al uso de *software* específico para el desarrollo y comercialización de nuestros libros**. Tampoco tendríamos que ver mermadas nuestras posibilidades de venta o de regalías según las condiciones de cada distribuidor.

A esto se suma la **ausencia de estándares en la distribución de los *ebooks***. No es fortuito ni inevitable que se tengan que hacer

procedimientos semejantes una y otra vez según la plataforma.

En realidad, todas las plataformas necesitan la misma información para poder colocar a la venta el libro. Si estas optaran directamente por una publicación estándar, como el EPUB, también se haría posible estandarizar su distribución.

Una idea para automatizar la distribución de *ebooks*

La idea es sencilla. Cada distribuidor requiere acceso al *ebook*, a la información para su venta y hasta algunas impresiones de pantalla. La clase de archivo que utiliza iTunes Producer para procesar toda esta información es una carpeta. Esta contiene, entre las capturas de pantalla y el *ebook*, un XML con toda la información necesaria para la venta.

Podría encontrarse un esquema XML estandarizado para esta información. Y, a partir de ahí, juntarlo con el libro y sus capturas de pantalla en una carpeta comprimida como ZIP. Con los accesos necesarios incluso se podría automatizar su subida mediante FTP o scp.

Con esto se abriría la posibilidad de poder **automatizar la distribución** y así tener pleno control de todos los procesos requeridos para la creación y subida de los *ebooks*.

Otra consecuencia sería la evasión de los nuevos intermediarios que funcionan como único distribuidor; cuyo trabajo consiste en mandar el libro a diversas plataformas.

¿Qué pasaría si no aceptamos los términos de los distribuidores y, en su lugar, hacemos presión para que se estandaricen y transparenten sus prácticas?

Sobre esta entrada

- Título original: «Dolores de cabeza frecuentes para un editor de *ebooks*».
- Título en el *blog* de Mariana: «Dolores de cabeza frecuentes para un editor de *ebooks*».
- Fecha de publicación: 28 de agosto del 2018.
- Redacción original.
- Revisión de Mariana.

Un giro digital en la edición y una propuesta sobre metodología editorial

¿Qué alternativas tenemos ante un panorama lleno de amargos desencantos entre la tradición editorial y las nuevas tecnologías de la información y la comunicación?

¿Qué otros planteamientos son posibles además de los comunes contratos entre entidades editoriales y empresas prestadores de servicios de *software*?

¿Cómo los editores estamos destinados a que la calidad ya siempre dependa de un intercambio económico por los conocimientos técnicos de terceros?

Origen de los problemas

La edición se hizo código. En otras palabras: el quehacer editorial ya no es disociable del desarrollo de *software*. El texto, como lo habíamos comprendido desde la invención de la imprenta, tiene nuevas dimensiones que aún no terminamos de entender del todo.

Una publicación no solo tiene tras de sí una tradición relativa a los procesos editoriales y gráficos. El traslado de la manera de trabajar a una computadora no es un cambio de tecnologías o de técnicas: es un **cambio de método**.

¿Cansado de que cada vez es más común exigir más formatos de una misma obra? Si es así y si para ti crear un formato más requiere recursos y tiempo adicionales, por desgracia, hay algo en tu metodología de trabajo que provoque este problema. No se trata

que una dificultad técnica, sino que es un **conflicto metodológico**.

Aunque parece sencillo empezar a concebir a la edición como un método, la reducción de estas tecnologías a su aspecto instrumental genera la idea de que la solución ha de yacer en quienes la usan y la desarrollan.

Es decir, para la mayoría de los editores la solución queda afuera de su esfera. Ahora quien programa es el que cuenta con la solución. Y esta dependencia no es agradable porque por décadas el editor no había tenido que recurrir a profesionales afuera de su campo para cumplir con sus tareas.

Subordinación exterior y ausencia de metodología

Eso no es lo peor, depender de profesionales externos a la edición provoca:

1. Una asistencia técnica a través de un pago de prestación de servicios que, por lo general, el editor no termina de comprender del todo, prestándose a diversos abusos.
2. Un desinterés del agente externo al cuidado editorial, ya que es común que trate las necesidades del editor como un problema solucionable a través de una vía técnica que, al menos para el caso de la lengua española, no cumple del todo con los estándares de calidad de su tradición.
3. Una amargura por parte de quienes editan, porque se percatan de que su calidad editorial empieza a estar condicionada por el control técnico de una metodología, al menos en parte, distinta a los procesos tradicionales de publicación.

Es posible atajar estos problemas si se buscan mecanismos que expliciten que **la edición es tradición, arte, profesión, pero también es método**.

Educación

La vía más evidente para comprender que la edición también es método es la educación: que quienes editan empiecen a aprender que gran parte de sus dificultades están relacionadas con:

- las metodologías que utilizan en su trabajo, y
- la ausencia de un panorama general sobre las posibilidades metodológicas para poder llevar a cabo su trabajo.

El sector editorial tiene un fuerte problema pedagógico. Por un lado, la educación continua que se ofrece a estos profesionales implica altos costos que no muchos pueden cubrir. ¿Cuántas veces hemos desistido de acudir a capacitaciones, talleres, seminarios, coloquios, ferias o estudios de grado porque no podemos pagarlo?

Sin embargo, el aspecto que, por el momento, quiero hacer hincapié es cómo **la aproximación, por lo general, no es integral**.

¿Cuántas veces hemos asistido a eventos para aprender a utilizar *x* o *y* programa, plataforma o lenguaje? ¿Cuántas veces hemos podido incrustar esos conocimientos en un panorama más general?

¿Cuántas veces lo que se nos enseña es *software* propietario que requiere de pago o suscripción? ¿Cuántas veces este *software* se nos vende como la panacea a nuestros problemas?

La formación que el editor necesita no es a base de *software*. Sí, al final quien edita se las ve con el *software*, pero eso solo es el

corolario de un proceso pedagógico que requiere de fundamentos teóricos y metodológicos.

El aprendizaje tecnológico no se basa en dominar un software

Un grave error en nuestra educación es suponer que «aprender a usar la computadora» es «aprender a usar *x* o *y software*». Comprobaremos una y otra vez que lo aprendido es o será obsoleto —hasta el punto de una capacitación sin fin que, más que aclarar, nos confunde—. Pero, con mayor lamento, nos daremos cuenta que al final carecemos de la seguridad de que sabemos utilizar nuestra única herramienta posible de trabajo.

Si estás cómodo con tu entorno de trabajo —probablemente paquetería de Microsoft o Adobe—, solo recuerda qué pasó cuando *la solución* y *el método* editorial se reducían a programas como PageMaker o QuarkXPress… ¿No fue grato tener que reaprender, cierto?

Una formación con un fuerte peso teórico y metodológico ayudaría a que los editores, sin importar los programas o los formatos iniciales o requeridos, puedan construir sus propias soluciones.

No se trata de aprender a usar *software* sino de tener un criterio sobre lo que se necesita para llegar del punto A —los archivos para la edición— al punto B —los formatos finales de la obra—.

Al final, el gran prejuicio con el que se batalla es que los programas y los formatos siempre son secundarios; que un formato no está asociado a un programa determinado. Incluso más radical aún: todo formato final es desechable.

Con una sólida preparación metodológica se rompe ese aire de complejidad y ese gran distanciamiento entre los programas

milagro y los formatos en boga, y lo que el editor realmente necesita.

La educación que necesitamos no es a modo de brincos, de programa en programa, para estar a la vanguardia. La educación que el editor requiere es una paulatina profundización en su método, cuyo reflejo técnico sea, precisamente, eso: una imagen, una muestra concreta de una metodología que no está encadenada a programas o formatos.

¿Soluciones o creación de nuevas necesidades?

La confianza puede llegar a ser de tal grado que cada vez que se anuncia una nueva plataforma o programa no es causa de desasosiego ni desconcierto.

No es que la novedad sea despreciada, sino que se hace fácil distinguir cuándo una herramienta en realidad viene a resolver una dificultad metodológica y cuánto esta solo es *bluff*: un programa que se declara como útil cuando no hace sino crear dependencia al editor sin jamás resolver sus dificultades —como el DRM—.

Por lo general, cuando un programa o plataforma nueva sale al mercado, se promociona como *la* herramienta que soluciona *x* problema en el ámbito de la edición. Sin embargo, muchas veces no resuelve gran cosa y, en su lugar, crea una dependencia tecnológica del editor hacia ella.

Si esta situación se observara desde un panorama más integral podría detectarse con más facilidad cuando estos programas o plataformas vienen a solucionar algo o aparecen para crear una nueva necesidad.

Laboratorios

El tiempo desperdiciado en saltar de programa en programa se convierte en horas que ayudan a profundizar en los aspectos metodológicos. Este escudriño puede evitar el mareo que de manera constante tiene el editor frente a las nuevas tecnologías de la información. Conforme el editor se empodere de su único medio de producción, se hace posible la investigación y la experimentación.

¿Suena raro, no es así? ¿Cuándo habíamos escuchado sobre investigación o experimentación editorial? La investigación relativa a la edición, por lo general, tiene un fuerte énfasis histórico o bibliófilo. La experimentación editorial, comúnmente, son ejercicios de prueba y error sobre programas o formatos empleados en la edición.

La idea de investigación que aquí se plantea tiene un **fuerte énfasis técnico y tecnológico.** Cuando quien edita empieza a tener un dominio sobre su trabajo se hace posible ahondar en la reflexión sobre lo que implica tener un único medio de producción.

Cómo y con qué se están haciendo las cosas

La investigación consistiría en pensar y repensar a cada instante cómo y con qué se están haciendo las cosas y qué son esas cosas editoriales.

¿Por qué, como «gremio», usamos tales programas y no otros? ¿Por qué este u otro formato es el más usado? ¿Cómo podemos automatizar las cosas? ¿Cómo es posible encontrar lazos entre la edición estandarizada y la que no lo es? En fin, ¿qué es esa cosa llamada *edición* cuando ha desaparecido su correlato análogo —el

papel y la página— del que siempre dependió?

Estas solo son unas cuantas preguntas para un marco teórico que es mucho más profundo que la constante discusión de los libros impresos contra los digitales. La investigación no tiene su interés principal en programas o formatos.

La investigación es en torno a cómo pensar la edición cuando todos sus recursos han sido acaparados por el código y cómo ver al texto cuando ya no solo es contenido, sino también estructura que no necesariamente tiene que ser *estructura textual* (!).

La experimentación tiene una intencionalidad semejante. No consiste en aplicar a prueba y error ciertas recetas. No es únicamente la implementación de otros ecosistemas editoriales ajenos a los que usamos.

Esta consiste en **crear nuevos medios o herramientas para aumentar la calidad editorial de nuestras publicaciones**. Pero no solo eso: experimentar es partir del texto incluso para derivar en productos no textuales (!).

Siendo editores nos cuesta mucho trabajo concebir la posibilidad de productos editoriales no textuales a partir de nuestra principal base de trabajo: el texto. Y seguirá sonando raro, porque el sector editorial recién está viendo que casarse con el texto no implica casarse con las publicaciones.

Como editor, ¿quieres publicar o quieres trabajar con el texto? Desde que la edición se hizo código esto deriva en actividades muy distintas, cuando durante siglos para la edición una era consecuencia de la otra.

Laboratorio de investigación editorial

Cuando se cuenta con un espacio para la investigación y

experimentación lo que se tiene es un laboratorio; en este caso, un laboratorio editorial.

El Taller de Edición Digital (TED) es un ejemplo. Con un fuerte énfasis pedagógico, el TED poco a poco va mostrando su verdadera cara: no es un taller para aprender a editar, sino un laboratorio editorial.

Pero, a pesar del optimismo, el TED es frágil. Sin todas las personas que organizan actividades paralelas al TED este taller no tendría vitalidad. ¿Es posible replicar espacios similares de laboratorio editoriales?

Son las universidades, las instituciones gubernamentales y las editoriales establecidas las que sin problemas podrían abrir laboratorios en su estructura.

En la edición no es común hablar de esta clase de espacios. Sin embargo, cuando quien edita es también el responsable de generar sus propias metodologías o herramientas los laboratorios se fundan como una infraestructura necesaria.

No podremos retomar las riendas de nuestra profesión hasta que no concibamos la necesidad de estos espacios. **El empoderamiento tecnológico y la independencia técnica del editor no basta con ser individual**.

En colectivo es posible armar espacios de discusión que van más allá de estrategias publicitarias o de añejos debates que solo exponen el grado de desconocimiento hacia los métodos ya necesarios para publicar.

En laboratorios es posible establecer esos espacios que la edición necesita. Ahí será posible que en sí misma satisfaga sus necesidades presentes y futuras. Su quehacer es una actividad traducida en publicaciones y texto, pero también en código y en elementos editoriales aún desconocidos.

¿El texto como un aspecto definitorio de la edición?

Editores, aún no completamos el mapamundi de nuestro mundo: estamos aún en una nueva etapa de descubrimientos. Aún nos encontramos en nuestro siglo XVI, con nuevos mundos por descubrir. Tenemos un gran camino por delante desde y más allá de lo que hemos conservado como tradición.

El sector y la tradición editoriales han cometido un error fundamental. Ambos han supuesto que lo más importante o más común para la edición ya ha sido descubierto y solo es preciso afinarlo o adaptarlo a nuevas tecnologías y técnicas.

La realidad es que la edición ha perdido su carácter definitorio al no estar ya anclada a los procesos análogos, el papel y la página que la vieron nacer. Desde hace décadas lo que llamamos «editar» y «publicar» requieren de una radical redefinición.

Esta será insuficiente si no se elimina de una vez por todas lo que se considera el núcleo de la edición: el texto. La noción común del texto como contenido, como un conjunto de líneas de texto, nunca ha sido, no es ni podrá ser el aspecto definitorio de la edición. Desde siempre la edición ha desbordado esta materialidad tan bruta y evidente, a la vez que protocolaria.

Apertura

Si la edición se hizo código, entonces también hay que *hackearla*. Bajo este panorama quien edita ya no es un usuario o consumidor de *software*. Es, asimismo, su arquitecto.

Para poder construir los espacios y las herramientas necesarias para esta profesión se requiere acceso al código. Hay que recordar

que una fundamental diferencia entre las tecnologías análogas y las digitales es que estas últimas precisan esconder su infraestructura para poder funcionar.

Con diferentes grados de complejidad, la maquinaria análoga siempre hacía posible observar su funcionamiento mientras se utilizaba. Sin embargo, el *hardware* con verlo no nos da una idea del *software* que está ejecutando. Pero tampoco el *software* permite observar su constitución al utilizarlo.

El *software* se ejecuta y el desarrollador es el que decide qué mostrarle al usuario a través de una interfaz. Si se pretende estudiar su funcionamiento no solo se requiere hacer pautas entre la ejecución y el análisis de las funciones, también es menester el acceso al código.

El sector editorial ha sido criticado constantemente en no ser transparente en la difusión de sus conocimientos y técnicas. La situación es más oscura si quien edita utiliza un *software* propietario cuyo único acceso es el ejecutable; es decir, la limitación de solo tener acceso al código máquina prácticamente inteligible para las personas.

Cambio de rumbo

La apertura se vuelve una necesidad para que el editor como programador pueda ahondar con la investigación y la experimentación editoriales. No es un capricho, como «gremio» no podremos mejorar si:

1. Nos concebimos solo como usuarios y no como arquitectos de *software*.
2. A nuestros pares los vemos como una competencia y a

nuestros *software* y metodologías como la diferencia que nos da mayor competitividad.
3. Nuestra labor de investigación y experimentación permanece cerrada.

Esto provoca que nos sentemos a pensar **cómo es posible una educación integral y la gestación de laboratorios** donde su financiamiento no depende directamente de la capacidad adquisitiva de sus asistentes o integrantes. La tarea no es fácil, pero el giro digital en la edición no es imposible: ya estamos en eso.

Sobre esta entrada

- Título original: «Un giro digital en la edición».
- Título en el *blog* de Mariana: «Un giro digital en la edición y una propuesta sobre metodología editorial».
- Fecha de publicación: 2 de octubre del 2018.
- Redacción original.
- Revisión de Mariana.

II. Libros electrónicos estándar

El formato de una publicación: cuello de botella en la edición

El formato es el principal punto de cuidado dentro de la edición, como ya lo he mencionado en otras entradas. Su importancia es tal que **la calidad de una publicación depende de su formato**.

Sin embargo, esta relevancia ha sido inversamente proporcional a su descuido: pocas son las personas que se preocupan por el formato antes de empezar con la edición.

El resultado es predecible dentro del proceso editorial: las dificultades en la producción de un libro aumentan —muchas veces de manera incontrolable— incluso hasta tener que volver a empezar de nuevo.

No solo en este sentido **el formato es un cuello de botella en la edición**, sino que su mal manejo en diversas ocasiones también es perceptible para el lector.

Calidad, control y apertura en la edición

Como punto de partida, **el formato es una cuestión de cuidado editorial**. Un buen formato nos permite mantener una alta calidad y control del proceso de edición. Del mismo modo, posibilita su apertura para utilizarlo en cualquier entorno, sea una publicación digital o impresa, te guste usar *software* privativo o libre, o prefieras trabajar al *modo Adobe* o no.

Además, un formato adecuado nos permite evitar dificultades comunes dentro de la edición:

- La estructura del contenido se mantiene, con independencia del archivo final —EPUB o PDF, por ejemplo—, por lo que se evita ambigüedad en los estilos, como la falta de uniformidad en encabezados, párrafos, etcétera.
- La conversión entre formatos es sencilla y no se presta a descrontrol, lo que permite poder maquetar el texto en el entorno que se prefiera, como puede ser LaTeX, Scribus, InDesign o Sigil, por mencionar algunos.
- El tiempo de publicación en diversos soportes —EPUB, MOBI, IBOOKS, PDF para impresión, etcétera— deja de ser proporcional a la cantidad de archivos deseados; incluso sus tiempos de producción disminuyen a segundos, como se demostró al editar el artículo «Historia de la edición digital».
- El contenido se conserva a lo largo del tiempo, sin importar el cambio de tendencias o que el *software* utilizado para hacer la publicación deje de tener soporte, evitándose ese dolor de cabeza —y derroche de recursos— de tener que *recuperar* información o, peor aún, tenerla que *rehacer*, como a muchas personas les ha pasado cuando han tenido que obtener información de archivos de PageMaker o QuarkXPress para usarlo en InDesign.

Pero ¿qué es un «formato adecuado»?

Los **requisitos mínimos para un formato óptimo** son que:

1. esté en un lenguaje de marcado;
2. el formateo sea previo a la edición o, por lo menos, se realice antes de crear el archivo final;

3. se opte por un formato abierto y estandarizado;
4. tenga un mecanismo de control de versiones, para evitar sobrescrituras indeseadas.

Con estos elementos podemos acercarnos un poco más al ideal de una publicación de alta calidad, ya que cabe resaltar que **el cuidado en la edición no se reduce al cuidado en el formato**. No obstante, un buen formato nos ayudará a solventar muchos de los errores que, por lo general, y de manera casi mágica aparecen cuando la obra ya fue publicada; por ejemplo, dedazos donde antes no los había.

El formato también nos permitirá un alto control, ya que al existir una estructura uniforme —donde es posible identificar los distintos niveles de encabezados, párrafos, bloques de cita, itálicas, negritas y más—, los cambios necesarios pueden hacerse sin muchos desvaríos e incluso de manera automatizada, por lo que la migración de un entorno a otro no es complicado.

No menos importante es que un formato con estas características está pensado para la longevidad. Un formato abierto y estandarizado permite que, a pesar de los constantes cambios en las tecnologías de la información o en el mundo de la edición, el contenido y los metadatos de nuestra obra puedan ser reutilizados según las pautas editoriales del momento.

Parecen obviedades, pero el vaivén de los profesionales de la edición en cuanto a adopción tecnológica ha evidenciado que, en general, **el sector editorial ha sido irresponsable al momento de tratar los contenidos editoriales**. Si la edición es el cuidado de la obra para su publicación, el formato es el principal elemento que definirá la calidad en la edición y su tiempo de producción.

Un trato adecuado en el formato exige un cambio de enfoque al momento de tratar el texto. Por tradición, la edición tenía una dimensión visual. Tanto el editor como el tipógrafo, el formador o el diseñador cuidaban del contenido acorde a lo que era visible. Una vez concluido su trabajo se publicaba el libro y el lector se encargaba de juzgar el libro.

Esta tradición en el contexto digital no se ha perdido y, lo más importante, *no debe de abandonarse*. Sin embargo **actualmente existen al menos tres dimensiones dentro de una publicación**, donde dos son necesarias para cualquier soporte:

1. **Dimensión visual o de diseño**: la capa perceptible para el público en general cuya importancia reside en la legibilidad.
2. **Dimensión estructural o de formato**: la capa oculta que da cohesión al diseño, por lo que es relevante para mantener el control en la edición.
3. **Dimensión funcional o de programación**: la capa que permite cambios dinámicos en el texto, la cual es propia de las publicaciones digitales y posibilita otras experiencias de lectura.

Lo ideal es que la dimensión estructural determine al menos las pautas mínimas de diseño. Al marcar dónde hay encabezados, párrafos, bloques o epígrafes, es posible aplicar estilos acorde al tipo de contenido.

Sin embargo, **en el modo habitual de crear una publicación se va de la dimensión visual para después determinar la dimensión estructural**. Si el diseño no es uniforme habrá errores de estructura que se trasladarán a futuras ediciones o a otros

soportes.

Pero eso no es lo más grave. Este modo de trabajo implica que quien crea la estructura no es el editor, sino su programa de edición que, como es de esperarse, carece de criterios editoriales al momento de crear *automáticamente* la estructura a partir del diseño.

Esta predilección por el diseño es la característica del enfoque WYSIWYG («lo que ves es lo que obtienes»). El enfoque que da preferencia a la estructura es otro: es el WYSIWYM («lo que ves es lo que quieres decir»). Los procesadores de texto, como Word, y los programas de maquetación, como InDesign, apuestan por el enfoque WYSIWYG. La adopción de este modelo se debe a que es más fácil de aprender y porque otorga mayor libertad al usuario.

Sin embargo, en un ambiente donde la calidad del formato es esencial, el enfoque WYSIWYM sobresale por establecer pautas de control que muchas veces se traduce en una curva de aprendizaje más larga. Aunque aprender este enfoque tome más tiempo, en un mediano o largo plazo, implicará un ahorro de tiempo, la supresión de erratas y, principalmente, un aumento en la calidad de las publicaciones.



Las dimensiones de una publicación.

Formatos recomendados

De manera concreta, ¿cuáles son los formatos siguen el enfoque WYSIWYM y al mismo tiempo tienen las características de optimización que buscamos? A continuación se hace un repaso somero sobre las posibilidades que contamos para mejorar el formato de nuestras publicaciones.

Markdown para el contenido

Los lenguajes de marcado ligero son la primera opción para quienes empiezan a trabajar con el enfoque WYSIWYM. Estos lenguajes sobresalen porque son fáciles de escribir y de leer, además de permitir su conversión sin problemas a otros formatos.

Un gran ejemplo de este tipo de lenguaje es Markdown. Con este recurso podemos empezar a marcar el contenido de nuestra publicación desde el mismo momento que leemos sobre su sintaxis básica. Un ejemplo es el siguiente:

```
# Encabezado 1
Esto es un párrafo con una *itálica*.
## Encabezado 2
Esto es un bloque de cita con una **negrita**.
```

La versatilidad de este formato ha ayudado a que se prescinda de procesadores de texto al momento de redactar un documento, como son las entradas que publico aquí. Como es de esperarse, Mariana prefiere otros formatos para editarlos, lo cual no es ningún problema porque gracias a Pandoc puedo generarle un documento de Word o cualquiera que sea de su preferencia.

Pero esta agilidad tiene un precio: no es posible dar formato a estructuras complejas, al menos no con la sintaxis básica de Markdown. Ejemplos de estas estructuras podrían ser epígrafes o párrafos franceses.

HTML para el contenido

Otro formato que nos permite abordar estructuras más complejas es HTML. Su flexibilidad es tal que es el formato empleado para los **libros electrónicos estándar**. Si bien su estructura no es tan cómoda de leer nos permite crear estructuras como la siguiente:

```
<h1>Encabezado 1</h1>
<p class="epigrafe">Esto es un epígrafe</p>
<p>Esto es un párrafo con una <em>itálica.</em></p>
<h2>Encabezado 2</h2>
<blockquote>Esto es un bloque de cita con una <stro
<p class="frances">Esto es un párrafo francés.</p>
```

Con el atributo de `class` podemos definir estilos particulares que luego se establecen en el diseño. Pero una publicación no solo es su contenido, sino también sus **metadatos**, lo cual no se soluciona de manera satisfactoria con Markdown o HTML.

YAML o JSON para los metadatos

Como Mariana lo mencionó en otra entrada, los metadatos permiten la catalogación de una obra. Un participante de un taller que impartimos resumió los metadatos como «el acta de nacimiento de un libro». Pienso que es una buena definición, ya

que sin metadatos no es posible identificar una publicación. Por esto no solo son importantes para los soportes digitales, sino también para los impresos, porque solo así es posible dar con ellos más allá de un encuentro fortuito en una librería.

Para un uso adecuado de los metadatos, mi recomendación es YAML o JSON. YAML es un formato muy flexible y fácil de usar, aunque puede fomentar una pérdida de control. En cambio, JSON es un formato que requiere de una mayor curva de aprendizaje pero con el fin de mantener un orden.

Un ejemplo de metadatos en YAML es:

```
---
# Generales
title: Sexo chilango
author: Braun, Mónica
publisher: Nieve de Chamoy
synopsis: Esta edición reúne todas las columnas…
category: Ficción, Narrativa
version: 2.0.0
```

Como se observa, su lectura y escritura es muy sencilla. Este es el motivo por el que en Perro Triste hemos optado por este formato para el manejo de los metadatos.

El mismo ejemplo en JSON nos da:

```
{
 "title": "Sexo chilango",
 "author": "Braun, Mónica",
 "publisher": "Nieve de Chamoy",
 "synopsis": "Esta edición reúne todas las columnas.
 "category": "Ficción, Narrativa",
 "version": "2.0.0"
}
```

La estructura no es tan complicada pero, debido a su sintaxis, la lectura y escritura se vuelven muy accidentadas. Además, habrá quien no le agrade manejar archivos distintos para el contenido y los metadatos, por lo que existe otra opción.

XML para el contenido y metadatos

El formato XML es un veterano en cuanto al cuidado de los contenidos. Su estructura está pensada para ser fácil de usar entre computadoras por lo que no resulta cómodo de leer o escribir. No obstante, su sintaxis no difiere mucho a la del HTML, como vemos en este ejemplo:

```
<?xml version="1.0" encoding="UTF-8" ?>
<publication>
```

```
  <metadata>
  <title>Sexo chilango</title>
  <author>Braun, Mónica</author>
  <publisher>Nieve de Chamoy</publisher>
  <synopsis>Esta edición reúne todas las columnas…</s
  <category>Ficción, Narrativa</category>
  <version>2.0.0</version>
  </metadata>
  <content>
  <h1>Encabezado 1</h1>
  <epigraph>Esto es un epígrafe</epigraph>
  <p>Esto es un párrafo con una <em>itálica.</em></p>
  <h2>Encabezado 2</h2>
  <blockquote>Esto es un bloque de cita con una <stro
  <hanging>Esto es un párrafo francés.</hanging>
  </content>
</publication>
```

Debido a que el XML es extendible —posibilita crear nuevas etiquetas —, permite la incorporación de estructuras complejas que involucren tanto los contenidos como los metadatos de una publicación.

Un gran ejemplo del uso de XML lo tenemos en los artículos académicos. Journal Article Tag Suite (JATS) es el formato XML estándar para muchos repositorios de artículos académicos como SciELO. Con esto se hace posible la creación de múltiples soportes de lectura al mismo tiempo que permite un uso flexible y longevo

de la publicación.

Otro ejemplo lo encontramos en la posibilidad de usar este formato para trabajar con InDesign, como ya se había explicado en otra entrada.

El formato y nuestra tradición cultural

La dicho anteriormente puede dejarnos la sensación de que el formato se reduce a una cuestión técnica en pos de un mejor cuidado editorial. No obstante, esto es solo el inicio cuyo punto de llegada es que **el formato es una cuestión cultural**.

La conservación y prolongación de nuestra herencia cultural ya es indisociable al mantenimiento de la infraestructura digital desde la cual creamos contenidos culturales, como son las publicaciones.

Desde una perspectiva individual, el formato toma relevancia cuando por la migración de *software* o de sistema operativo nos vemos imposibilitados de abrir un archivo que necesitamos.

En un grupo de trabajo, el formato adquiere importancia cuando entre los mismos compañeros es imposible realizar una tarea, ya que cierto archivo no es compatible o su formato se rompe al usarlo en otra computadora. En una editorial, el formato es significativo cuando resulta difícil recuperar la información almacenada en archivos privativos antiguos.

Si continuamos escalando las dificultades surgidas a partir de un mal manejo en el formato de los documentos tenemos casos alarmantes en donde parte de nuestra herencia cultural queda «secuestrada».

Es decir, **la información está «ahí» pero no es accesible y su recuperación es incosteable** para diversas instituciones. Casos

lamentables de este problema lo podemos ver en archivos, bibliotecas y repositorios públicos cuyas consultas quedan obstaculizadas por un «mero problema técnico».

En el trabajo del día a día, el formato es una cuestión técnica que, por lo general, se opta por la manera más sencilla de llevarlo a cabo, sin consideración alguna sobre lo que esto puede implicar para el futuro. Sin embargo, en una dimensión más general y a largo plazo, la calidad y apertura en el formato determinará qué tan accesible y flexible será su uso para los siguientes años o décadas.

Dada la complejidad del asunto, **es comprensible que el sector editorial esté harto del *software* milagro**, de aquellas «novedades» tecnológicas que supuestamente facilitan su trabajo, pero que al final terminan por complicarlo. A nadie le agrada aprender a usar programas una y otra vez, mucho menos recuperar la información para tener la posibilidad de usarla.

En este sentido, quizá **el sector editorial debería empezar a plantearse con seriedad la manera de manejar los contenidos** con una perspectiva a largo plazo y con independencia al *software* en boga. Tal vez así sea posible mejorar la calidad editorial sin el temor de siempre volver a empezar de nuevo…

Sobre esta entrada

- Título original: «El formato de una publicación: cuello de botella en la edición».
- Título en el *blog* de Mariana: «El formato de una publicación: cuello de botella en la edición».
- Fecha de publicación: 2 de mayo del 2017.
- Redacción original.

- Revisión de Mariana.

Como lo prometido es deuda, en la entrada «Edición digital como edición desde cero» se mencionó que se verían con mayor detenimiento tres de los métodos que se compararon al momento de crear un EPUB de la edición del Proyecto Gutenberg del *Don Quijote*.

Para descansar *un poco* de las nociones relacionadas al código, **el primer método que abordaremos es la creación de un EPUB con InDesign**.

El desarrollo de EPUB con InDesign puede hacerse de muchas maneras. Sin embargo, aquí **nos enfocaremos en el proceso que ofrece más control, menos tiempo y mejor calidad**. A partir de un XML y una hoja de estilo CSS crearemos un EPUB de una manera muy sencilla, al contrario a lo que se podría pensar en un principio.

1. Preparación de los archivos

Antes de abrir InDesign **necesitamos tres archivos** para nuestro EPUB: **el XML, el CSS y la portada**. Solo el XML es necesario, ya que en este estará todo el contenido de nuestro libro. Más adelante explicaremos las particularidades de este formato.

La portada puede ser en cualquier formato de imagen y de cualquier tamaño. En mi opinión, lo más recomendable es que esté en JPG, ya que este formato tiende a pesar menos.

En cuanto al tamaño, en Nieve de Chamoy usamos una **altura no mayor a 2048 pixeles** por el único hecho de que en el iPad la

portada se muestra en pantalla completa por algunos segundos, tiempo suficiente para que el usuario note algún *pixeleo* en la imagen. Si no se piensa distribuir para iOS, una altura no mayor a 1024 pixeles tendría que ser suficiente.

Estas pequeñas particularidades de la portada es un detalle importante ya que, por lo general, su peso puede representar **entre el 25 y 50 por ciento del tamaño final del EPUB**. Además, recuérdese que, entre más ligero el libro, el costo de envío con Amazon será menor (véase el apartado C de este enlace), o bien, la descarga será más rápida y ocupará menos espacio en el dispositivo del usuario.

Si no se desea agregar una imagen, InDesign puede generar una portada a partir de la primera página de nuestro libro o simplemente no añadirla. Esta última opción se desaconseja, ya que la vista por defecto de la estantería de los *ereaders* solo muestra la portada del libro.

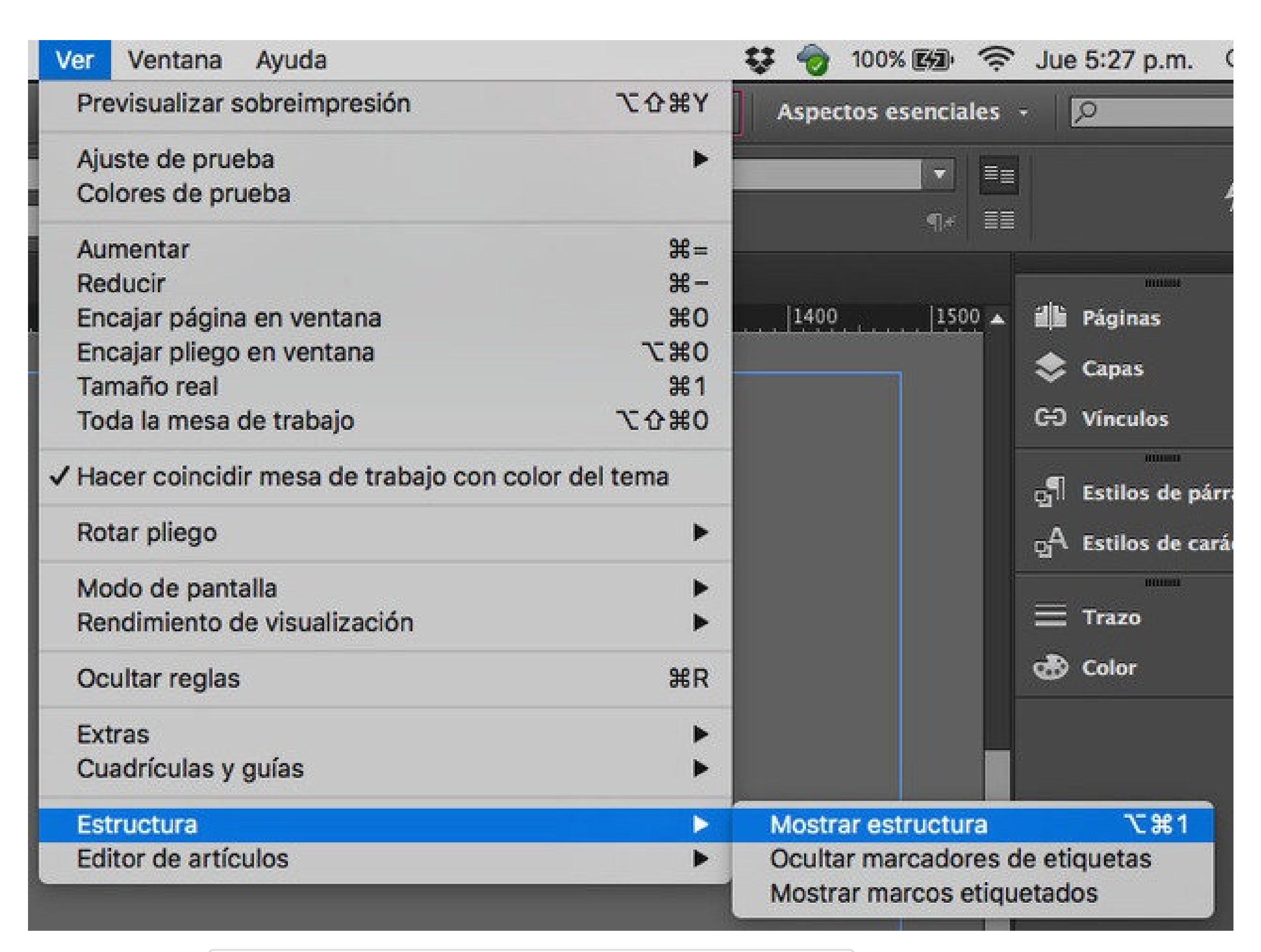
La **hoja de estilos CSS** no es necesaria; sin embargo, su implementación nos **permite un mejor y mayor control en el diseño de nuestro EPUB**. Si el editor desconoce por completo este lenguaje de marcado, la recomendación es que un diseñador *web* genere una plantilla que se pueda utilizar en varios de libros.

Si no es posible la ayuda de un diseñador *web*, sin ningún problema puede utilizar la plantilla usada para este EPUB, la cual es cortesía de Perro Triste. El CSS puede verse aquí o descargarse mediante el enlace que está al final de este artículo.

El **archivo XML** es un formato que **contiene una serie de etiquetas dentro de las cuales se agrega el contenido de manera semántica**. Es decir, si en nuestro libro tenemos un párrafo, este estaría rodeado de unas etiquetas `<p>`, por poner un ejemplo.

Para generar este archivo existen dos medios generales: convirtiendo el archivo de texto original, por lo general un `.doc` o `.docx`, a XML o etiquetar directamente el texto que ya está presente en InDesign.

Esta última opción se desaconseja si el libro solo tendrá una salida para EPUB, ya que por lo general implica una pérdida de control sobre el contenido semántico del libro. Para agregar el texto ya existente a una estructura XML primero hay que ir a `Ver > Estructura > Mostrar estructura`.



Ubicación de `Ver > Estructura > Mostrar estructura`.

Con esto habilitaremos la visualización de la estructura XML del documento. Para agregar el texto es necesario ir al menú de esta nueva ventana y hacer clic en `Añadir elementos no etiquetados`.



Ubicación de `Añadir elementos no etiquetados`.

Estructurar de esta forma puede llevarnos minutos o incluso horas.

Para mayor información al respecto, lo mejor es comenzar con el sitio oficial de soporte de Adobe. En este artículo explica este proceso de manera muy completa.

En este ejercicio **estaremos importando un archivo XML, en lugar de generar su estructura desde InDesign**. El motivo de esto es que, previamente, a través de un «archivo madre» la obra es vaciada a etiquetas para después convertirla según las herramientas que se emplearán para crear el libro.

El «archivo madre» tiende a convertirse en alguno de estos formatos: HTML o XHTML para EPUB «desde cero» o con Sigil, TeX para PDF a través de LaTeX o XML para InDesign, independientemente de si se trate de un impreso (PDF) o un *ebook* (EPUB). Este flujo de trabajo supone que:

1. una obra tendrá distintos formatos, como pueden ser EPUB, PDF, MOBI o en línea;
2. el editor es el único responsable de asignar semánticamente el contenido, evitando que este trabajo sea realizado por el desarrollador o diseñador que, por lo general, no tienen un amplio conocimiento sobre la obra;
3. InDesign solo forma una parte del proceso de producción del libro en lugar de ser el único flujo de trabajo para la edición digital.

Esta metodología se conoce como *single source publishing*. Para una descripción más extensa, recomendamos el artículo «Historia de la edición digital» publicado en este sitio.

Una vez que el «archivo madre» se convierte a formato HTML, la generación del XML es muy sencilla. Al HTML solo se le tiene que eliminar todo el código que no forme parte del `body`, así como

guardar este archivo con la extensión `.xml`.

Si existen etiquetas con alguna clase se recomienda cambiar el nombre de la etiqueta por el nombre de la clase, para así tener la posibilidad de distinguirla una vez que estemos en InDesign. Por ejemplo, si en el HTML tenemos `<p class="pre">Un párrafo con clase «pre»</p>`, habría que cambiarlo por `<pre>Un párrafo con clase «pre»</pre>`.

Una vez hechas estas modificaciones, las cuales pueden automatizarse mediante `Buscar todo`, RegEx o `Reemplazar todo`, la estructura será semejante a la imagen que se muestra a continuación, donde los puntos suspensivos representan más etiquetas que no se muestran por cuestiones de espacio.

```
<body>
    <pre>The Project Gutenberg EBook of Don
    <pre>This eBook is for the use of anyon
    <pre>Title: Don Quijote</pre>
    <pre>Author: Miguel de Cervantes Saaved
    <pre>Posting Date: April 27, 2010 [EBoo
    <pre>Release Date: December, 1999</pre>
    <pre>Language: Spanish</pre>
    <pre>Character set encoding: ISO-8859-1
    <pre>*** START OF THIS PROJECT GUTENBER
    <pre>Produced by an anonymous Project G
    ...
</body>
```

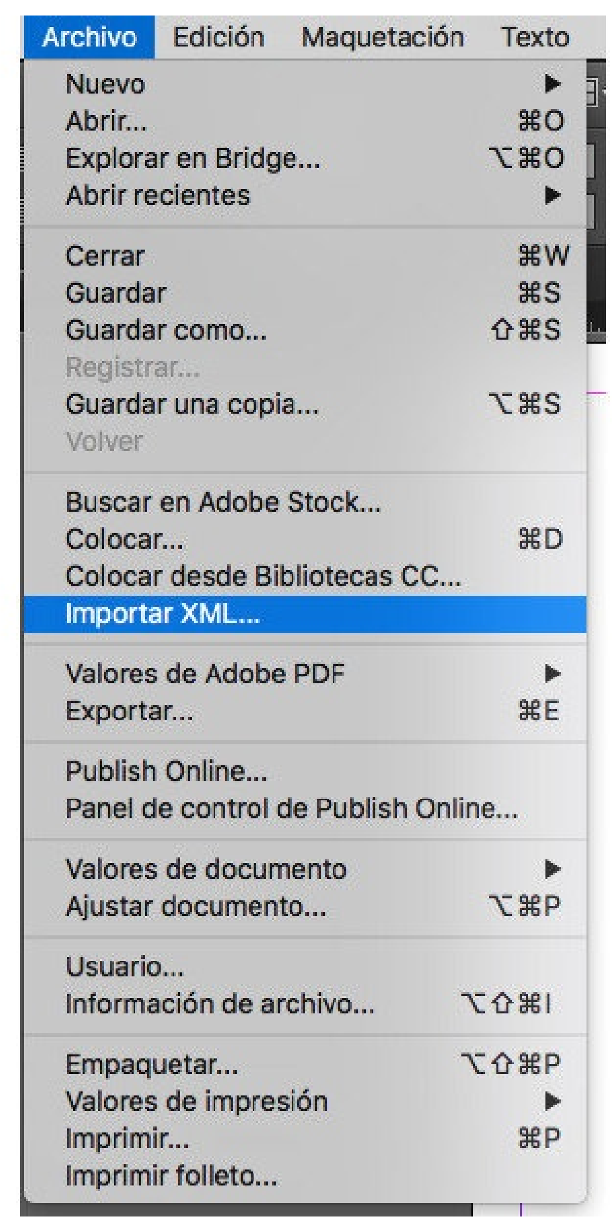
Ejemplo de correcciones realizadas.

Con esto ya tenemos listo un archivo que podremos importar a InDesign.

2. Importación del archivo XML

Si se generó la estructura a partir de un texto ya existente en InDesign, como se mencionó en el punto anterior, este paso no es necesario, porque ya se habrá creado la estructura, aunque aún falte darle el orden deseado.

Si no fue así, y solo se desea crear un EPUB con InDesign, lo más recomendado es partir de un documento nuevo de InDesign cuyas características son indiferentes. Como sea, para incorporar el archivo XML es necesario ir a `Archivo > Importar XML…`.

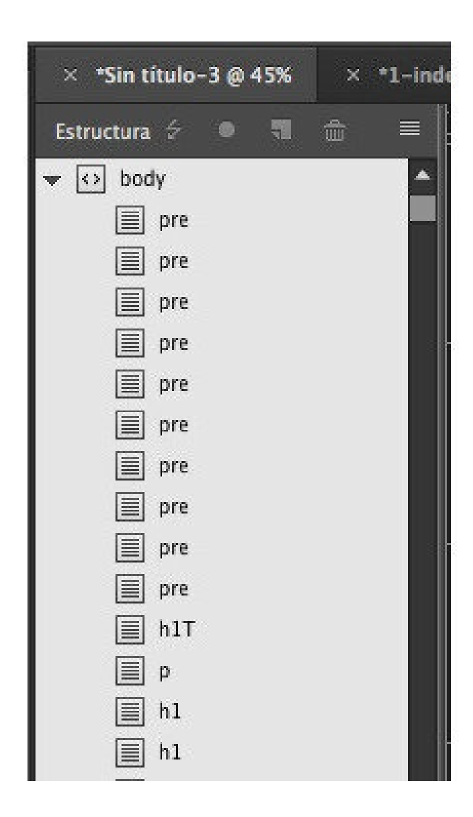


Ubicación de `Archivo > Importar XML…`.

La ventana que se abre da dos opciones: `Combinar contenido` o `Anexar contenido`. Cuando se trata de la primera importación cualquiera de las opciones es indiferente. Sin embargo, si previamente existe una estructura XML la primera opción nos sustituirá el contenido —ideal cuando se quiere agregar una versión más reciente—, mientras que la segunda lo añadirá al final

—ideal si la obra no está en un solo archivo XML—.

Por último, seleccionamos nuestro contenido sin necesidad de mostrar las opciones de importación, ya que esta opción es solo para usuarios avanzados.



Importación del XML.

De esta manera, se importará el XML y automáticamente nos mostrará la ventana con su estructura. Con esto ya podemos pasar al siguiente paso.

3. Creación y asociación de los estilos de párrafo

Ahora es necesario crear una asociación de las etiquetas XML con los estilos de párrafo. Este paso solo es necesario si se desea tener un control en el diseño del EPUB a través de hojas de estilos CSS. Si los

estilos se manejan de manera directa o se quieren preservar los actuales, puede omitirse este paso y pasar al siguiente paso.

Nosotros siempre recomendamos el uso de CSS para el EPUB, independientemente de si el libro ya tiene un diseño para la impresión, porque:

1. existe un gran control sobre el diseño del *ebook* y se evita tener que optimizar mediante el uso de otras herramientas, como puede ser Sigil;
2. disminuye el peso del libro ya que normalmente el diseño creado con InDesign carece de uniformidad y crea varias líneas de código que pueden ser conflictivas;
3. si se necesita modificar algo en el diseño es más fácil la comprensión del código presente en una plantilla CSS que el creado por InDesign, y
4. si se trata de una colección, es posible unificar el diseño de todas las obras que la componen.

Lo primero que debe hacerse es **crear los estilos de párrafo**, **uno por cada tipo de etiqueta** que contiene nuestro documento. Se recomienda que el nombre del estilo sea el mismo que el de la etiqueta para su fácil y automática asociación. Por ejemplo, si en el XML tenemos una etiqueta `<pre>`, el nombre del estilo de párrafo sería `pre`.

En esta obra, *Don Quijote*, tenemos once estilos. De esos once seis son de encabezados, que corresponden a cinco niveles distintos más uno reservado para el título. Los cinco restantes son tipos de párrafo, como son los cuerpo de texto, centrado, alineado a la derecha, cuerpo preformateado para la página legal y los versos.

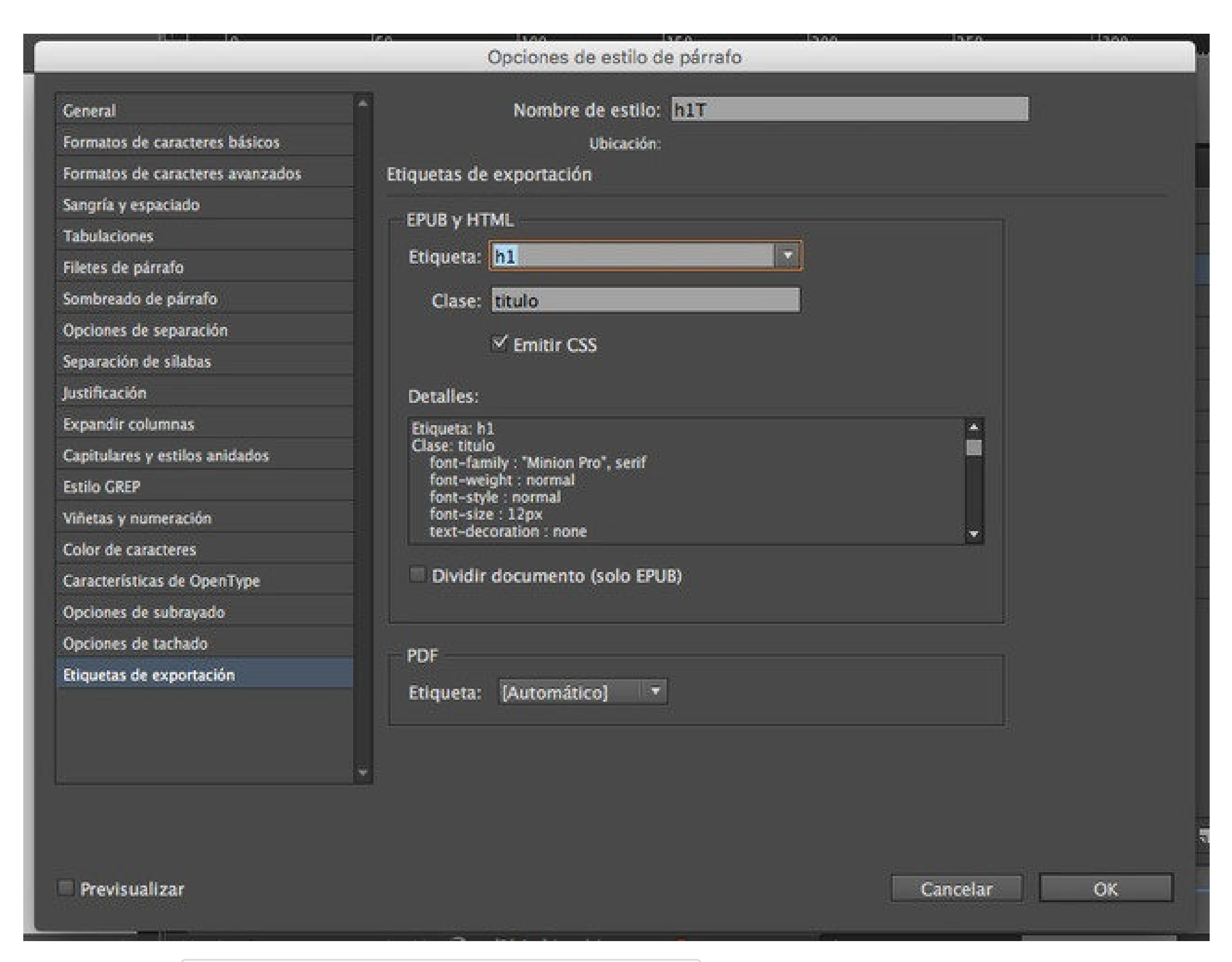


Ventana de `Estilos de párrafo`.

De las opciones disponibles para los estilos de párrafo, para el EPUB solo nos interesa el apartado `Etiquetas de exportación`. Aquí tenemos que indicar qué etiqueta se creará en el EPUB para sustituir el estilo de párrafo, así como indicar si esta etiqueta tendrá alguna clase CSS asociada.

Por ejemplo, el estilo de párrafo `h1T` es para el título, el cual, según la hoja de estilos CSS que vamos a incorporar, este debe de estar dentro de una etiqueta `h1` con la clase `titulo` (para evitar conflictos, evítense las tildes). Otros ejemplos son: para el

estilo de párrafo `pre`, una etiqueta `p` (párrafo) con la clase `pre`; para el estilo de párrafo `h1`, una etiqueta `h1`, sin necesidad de agregar alguna clase CSS.



Ventana de `Opciones de estilos de párrafo`.

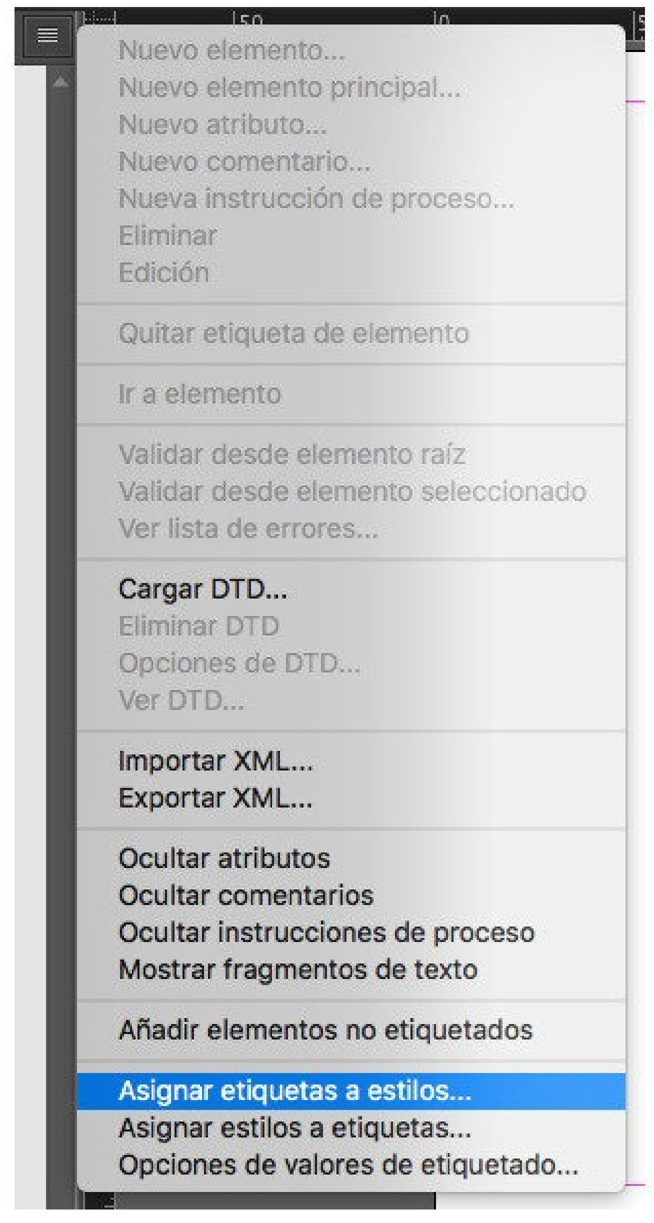
Si este paso no se realiza, InDesign no sabrá qué etiqueta HTML corresponde a los estilos de párrafo y estructura XML de nuestro documento. Además, de esta manera InDesign podrá ignorar toda la configuración tipográfica del párrafo *únicamente* cuando cree un EPUB, por lo que es posible **manejar el estilo para impresión con la seguridad de que este no interferirá con el diseño del EPUB.**

Ahora es necesario **asociar etiquetas a estilos y estilos a etiquetas**. Parece redundante, pero son dos opciones distintas que ofrece InDesign al momento de trabajar con estilos de párrafo y estructura XML. En el primer caso se indica que, por ejemplo, a toda etiqueta `<pre>` le corresponde el estilo de párrafo `pre`.

En la otra opción se indicaría que, continuando con el ejemplo, toda clase de párrafo `pre` ha de ser una etiqueta `<pre>` por lo que, de existir un párrafo con esta clase pero con una etiqueta XML distinta, supongamos que `<cen>`, automáticamente se cambiará a la etiqueta asociada, `<pre>` en este caso.

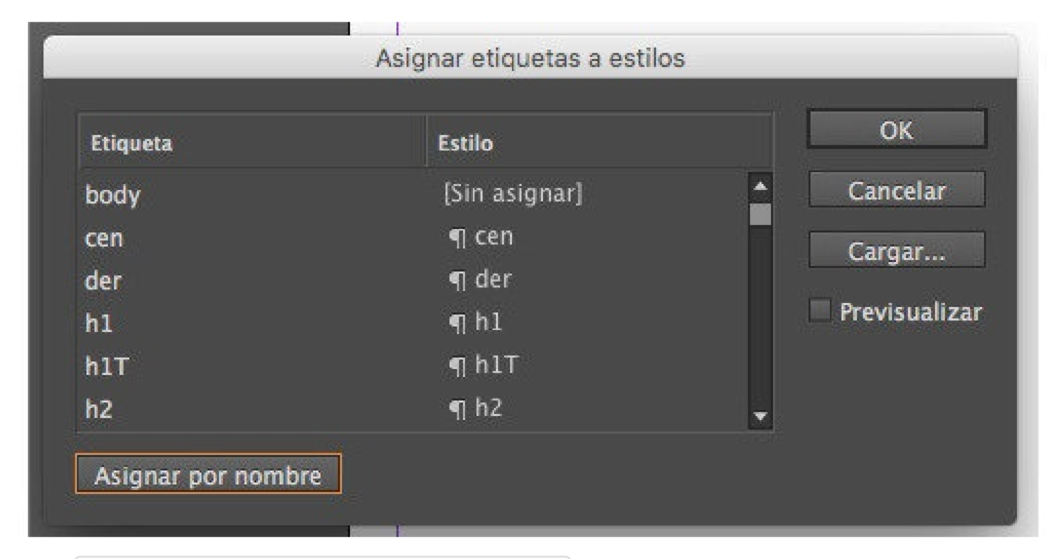
Esta doble asociación es un tanto confusa; sin embargo, lo único que tenemos que tener en cuenta es que, si no se realiza, el EPUB creado no tomará en cuenta la hoja CSS que se añadirá.

Entonces, en el menú de la ventana de la estructura XML nos vamos a `Asignar etiquetas a estilos…`.



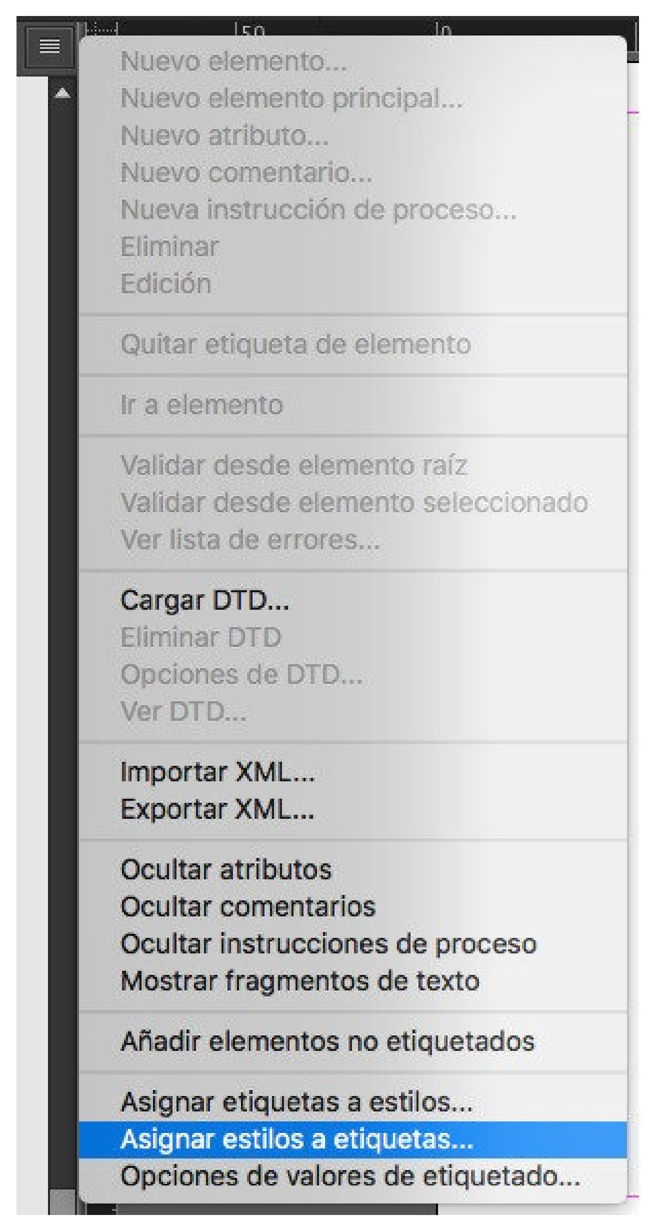
Ubicación de `Asignar etiquetas a estilos…`.

Si nuestras etiquetas tienen el mismo nombre que los estilos de párrafo, la asociación se puede automatizar al hacer clic en `Asignar por nombre`.



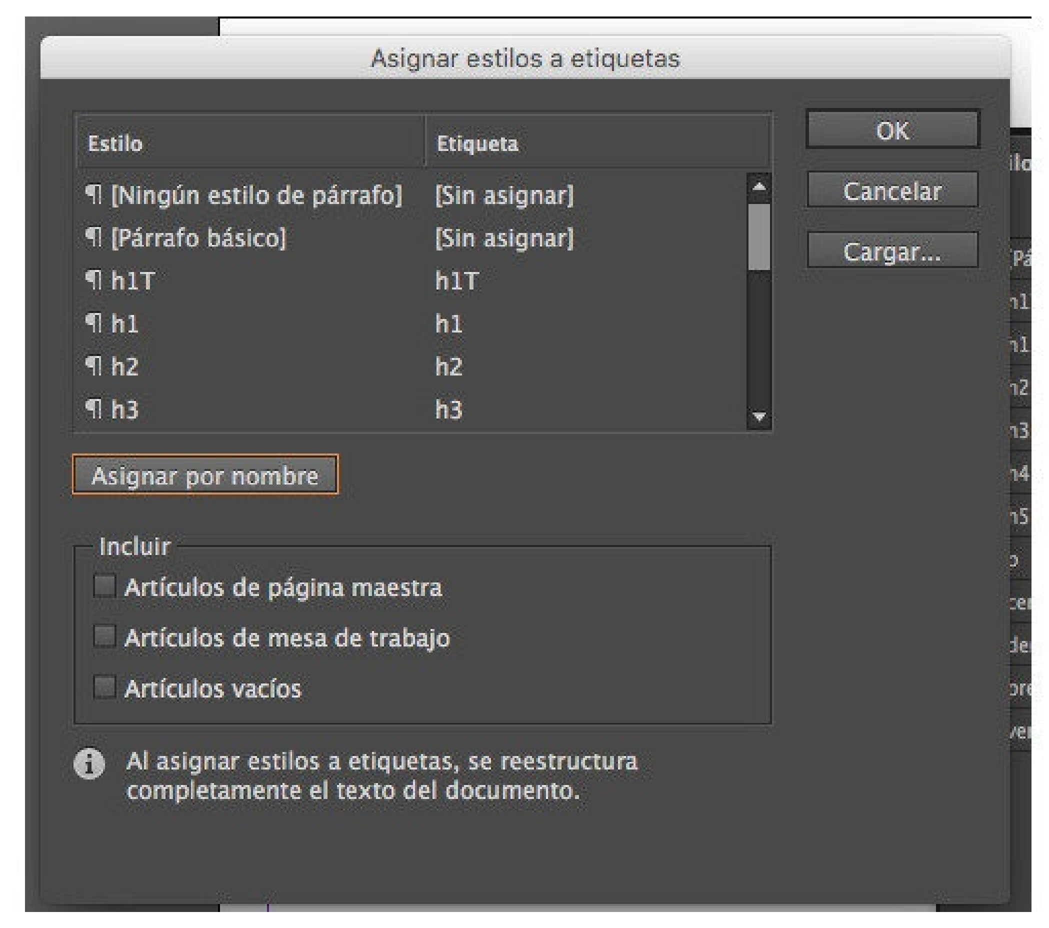
Ventana de `Asignar etiquetas a estilos`.

En el mismo menú nos vamos a `Asignar estilos a etiquetas…`. Nótese que es indistinto el orden por el que se asocian los elementos, bien se pueden asignar antes los estilos a las etiquetas. Lo relevante es que se lleve a cabo esta doble asignación.



Ubicación de `Asignar estilos a etiquetas…`.

Ya solo es cuestión de realizar la asociación. Si nuestros estilos de párrafo tienen el mismo nombre que las etiquetas, este proceso se puede automatizar al presionar sobre `Asignar por nombre`. Una vez hecho esto, podemos pasar a verter el texto.



Ventana de `Asignar estilos a etiquetas`.

4. Incrustación del texto y creación de la tabla de contenidos

Si el texto aún no está en una página, solo es necesario arrastrar la etiqueta `body` a algún espacio de esta. Si el libro solo se exportará en formato EPUB, no es necesario preocuparnos por la caja ni los números de página, por lo que no hay inconveniente con tener texto desbordado o que la página sea estéticamente desagradable. Solo **se busca que InDesign identifique que ahí hay un texto para convertirlo en EPUB**.

The Project Gutenberg EBook of Don Quijote, by Miguel de Cervantes Saavedra

This eBook is for the use of anyone anywhere at no cost and with almost no restrictions whatsoever. You may copy it, give it away or re-use it under the terms of the Project Gutenberg License included with this eBook or online at www.gutenberg.net

Title: Don Quijote

Author: Miguel de Cervantes Saavedra

Posting Date: April 27, 2010 [EBook #2000]

Release Date: December, 1999

Language: Spanish

Character set encoding: ISO-8859-1

*** START OF THIS PROJECT GUTENBERG EBOOK DON QUIJOTE ***

Produced by an anonymous Project Gutenberg volunteer. Text file corrections and new HTML file by Joaquin Cuenca Abela.

El ingenioso hidalgo don Quijote de la Mancha

Miguel de Cervantes Saavedra

Primera parte

Tasa

Yo, Juan Gallo de Andrada, escribano de Cámara del Rey nuestro señor, de los que residen en su Consejo, certifico y doy fe que, habiendo visto por los señores dél un libro intitulado El ingenioso hidalgo de la Mancha, compuesto por Miguel de Cervantes Saavedra, tasaron cada pliego del dicho libro a tres maravedís y medio; el cual tiene ochenta y tres pliegos, que al dicho precio monta el dicho libro docientos y noventa maravedís y medio, en que se ha de vender en papel; y dieron licencia para que a este precio se pueda vender, y mandaron que esta tasa se ponga al principio del dicho libro, y no se pueda vender sin ella. Y, para que dello conste, di la presente en Valladolid, a veinte días del mes de deciembre de mil y seiscientos y cuatro años.

Juan Gallo de Andrada.

Testimonio de las erratas

Este libro no tiene cosa digna que no corresponda a su original; en testimonio de lo haber correcto, di esta fee. En el Colegio de la Madre de Dios de los Teólogos de la Universidad de Alcalá, en primero de diciembre de 1604 años.

El licenciado Francisco Murcia de la Llana.

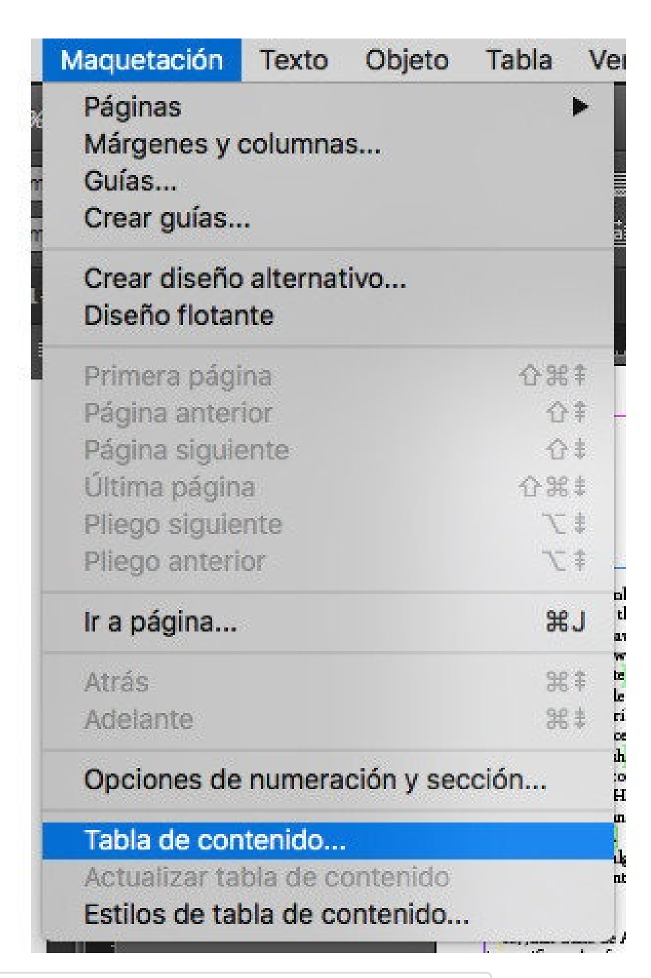
El Rey

Por cuanto por parte de vos, Miguel de Cervantes, nos fue fecha relación que habíades compuesto un libro intitulado El ingenioso hidalgo de la Mancha, el cual os había costado mucho trabajo y era muy útil y provechoso, nos pedistes y suplicastes os mandásemos dar licencia y facultad para le poder imprimir, y previlegio por el tiempo que fuésemos servidos, o como la nuestra merced fuese; lo cual visto por los del nuestro Consejo, por cuanto en el dicho libro se hicieron las diligencias que la premática últimamente por nos fecha sobre la impresión de los libros dispone, fue acordado que debíamos mandar dar esta nuestra cédula para vos, en la dicha razón; y nos tuvímoslo por bien. Por la cual, por os hacer bien y merced, os damos licencia y facultad para que vos, o la persona que vuestro poder hubiere, y no otra alguna, podáis imprimir el dicho libro, intitulado El ingenioso hidalgo de la Mancha, que desuso se hace mención, en todos estos nuestros reinos de Castilla, por tiempo y espacio de diez años, que corran y se cuenten desde el dicho día de la data desta nuestra cédula; so pena que la persona o personas que, sin tener vuestro poder, lo imprimiere o vendiere, o hiciere imprimir o vender, por el mesmo caso pierda la impresión que hiciere, con los moldes y aparejos della; y más, incurra en pena de cincuenta mil maravedís cada vez que lo contrario hiciere. La cual dicha pena sea la tercia parte para la persona que lo acusare, y la otra tercia parte para nuestra Cámara, y la otra tercia parte para el juez que lo sentenciare. Con tanto que todas las veces que hubiéredes de hacer imprimir el dicho libro, durante el tiempo de los dichos diez años, le traigáis al nuestro Consejo, juntamente con el original que en él fue visto, que va rubricado cada plana y firmado al fin dél de Juan Gallo de Andrada, nuestro Escribano de Cámara, de los que en él residen, para saber si la dicha impresión está conforme el original; o traigáis fe en pública forma de cómo por corretor nombrado por nuestro mandado, se vio y corrigió la dicha impresión por el original, y se imprimió conforme a él, y quedan impresas las erratas por él apuntadas, para cada un

Muestra de caracteres no imprimibles.

Como se podrá notar, el texto mostrará corchetes no imprimibles y de colores. Este es un apoyo visual de InDesign para poder distinguir el tipo de etiqueta XML de cada párrafo.

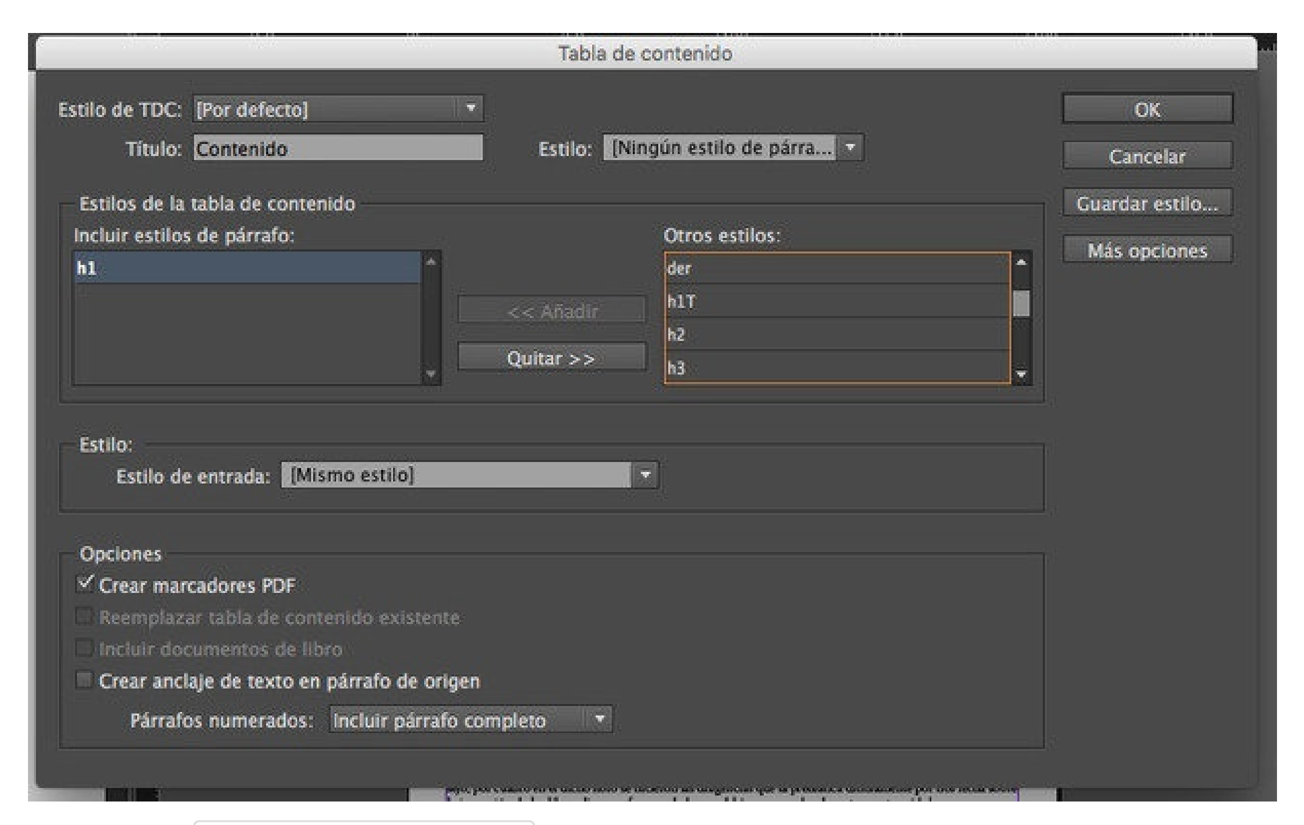
Para la **creación de la tabla de contenidos** hay que ir a `Maquetación > Tabla de contenido…`.



Ubicación de `Maquetación > Tabla de contenido…`.

La única opción que nos interesa en la ventana que se abre es el apartado de `Estilos de la tabla de contenido` en donde vamos a incluir los estilos de párrafo que queremos que formen

parte de nuestro índice. En este libro solo nos interesa que la tabla de contenido contenga los encabezados de primera jerarquía, excepto el título, por lo que solo agregamos el estilo de párrafo `h1`.



Ventana de `Tabla de contenido`.

Ya solo es necesario agregar la tabla de contenidos al documento. De nueva cuenta, para un libro que solo será EPUB no es necesario colocarlo en un lugar en especial. Incluso es posible dejarlo en un área exterior a la página, como se muestra en la siguiente imagen. Lo único que queremos explicitar a InDesign es que este documento contiene un índice que ha de incorporarse al EPUB.

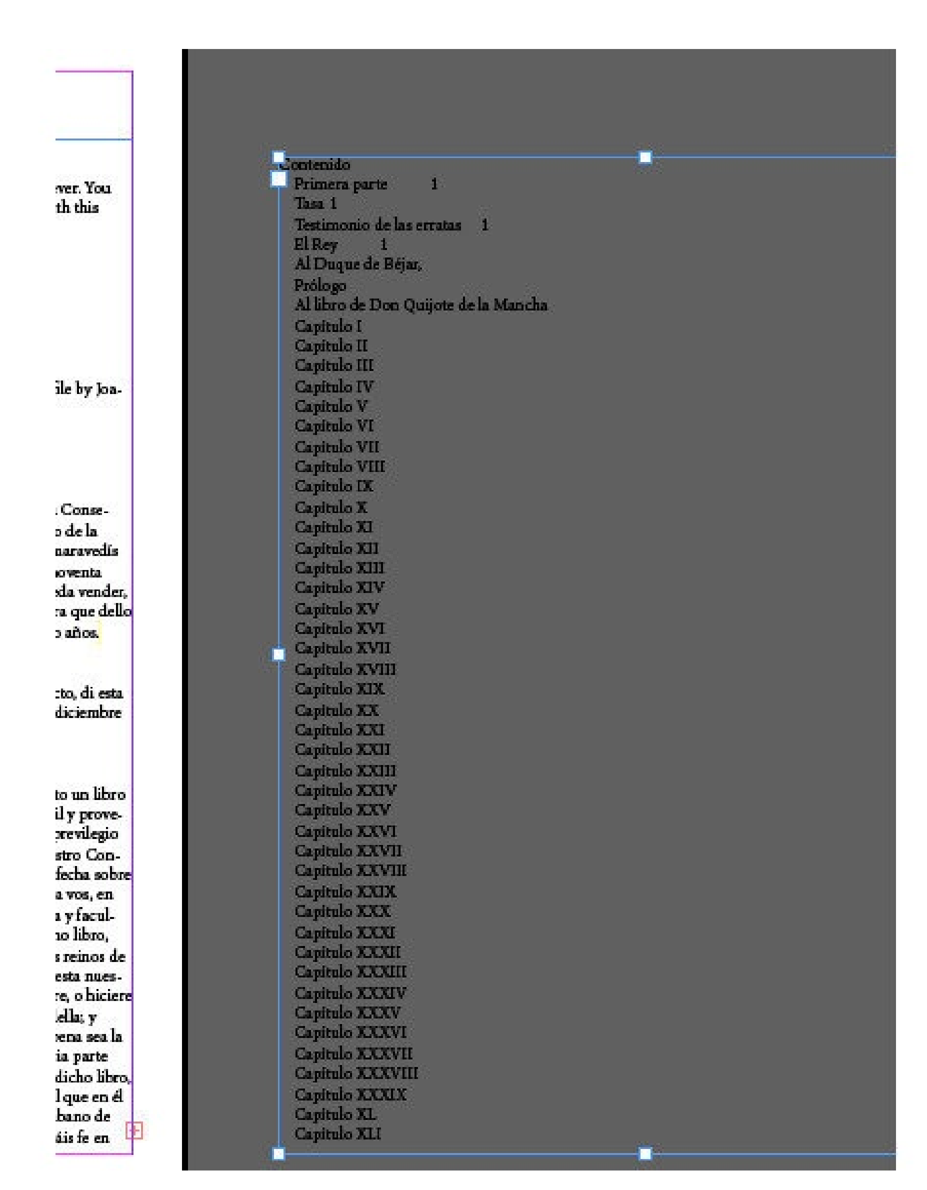
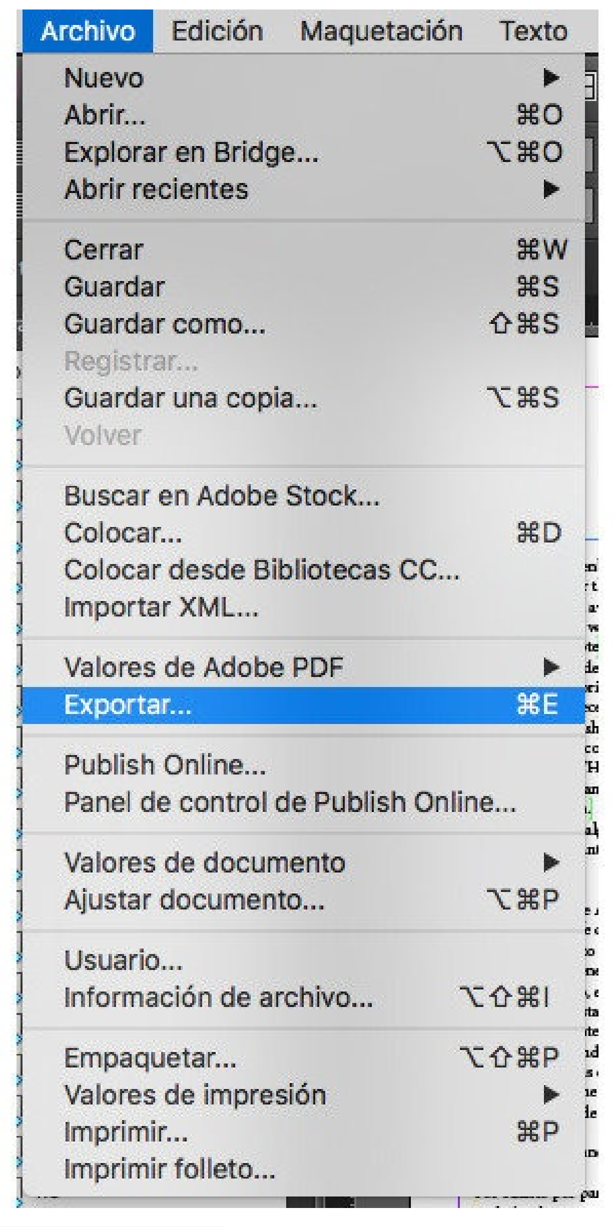


Tabla de contenidos creada.

5. Exportación de EPUB con InDesign

Por fin tenemos todo lo necesario para crear un EPUB con InDesign. A continuación nos vamos a `Archivo > Exportar…`.



Ubicación de `Archivo > Exportar…`.

Las características de la obra nos permiten que no nos preocupemos por un diseño fijo, el cual suele estar destinado para libros de apoyo didáctico o para niños. Entonces, en el formato elegiremos `EPUB (ajustable)`, también conocido como de **«diseño fluido» o «diseño responsivo»**, ya que se adaptará al tamaño de la pantalla de manera automática.

Formato: EPUB (ajustable)
Ocultar extensión | Carpeta nueva | Cancelar | Guardar

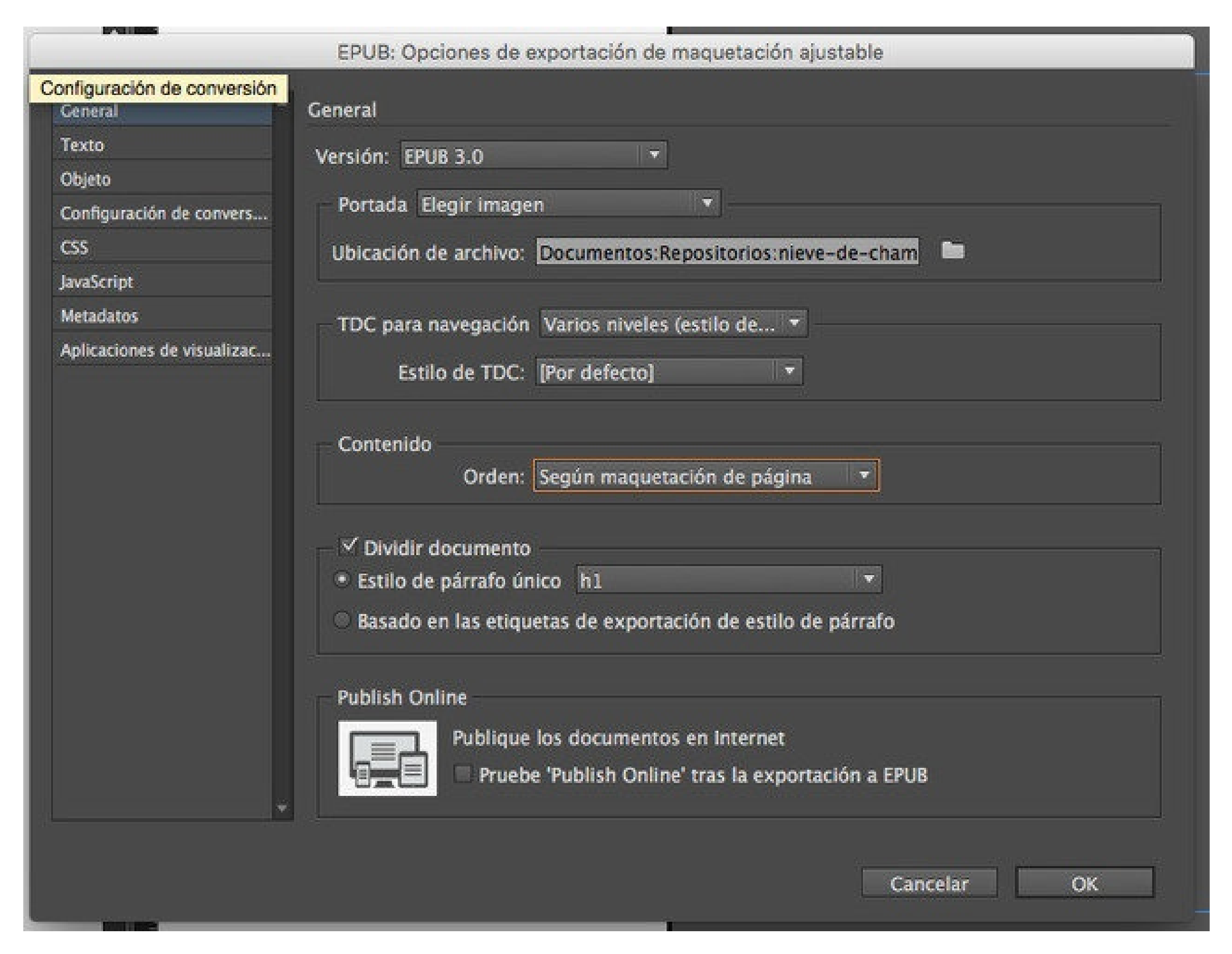
Formato de exportación.

Se abrirá una nueva ventana para dar los últimos ajustes al EPUB. La mayoría de los casos aquí solo tenemos que prestar atención a tres apartados. (Para una descripción sobre las opciones que no se verán aquí, puede echarse un vistazo aquí).

El primer apartado es `General`. Aquí se definirán las características globales del EPUB.

- Seleccionaremos la versión EPUB. Recomendamos la versión `3.0` por sus mayores posibilidades y flexibilidad. (Para una información detallada sobre las diferencias entre la versión `2.0.1` y la `3.0`, visítese este enlace).
- Elección de la portada. En este caso añadiremos una imagen externa, por lo que seleccionaremos `Elegir imagen`.
- Para que nuestra tabla de contenidos sea añadida en la sección `TDC para navegación` hay que seleccionar la opción `Varios niveles`.
- En `Contenido` seleccionaremos la opción `Según maquetación de página`. Así nos aseguraremos que el texto se mostrará tal cual como está vertido en las páginas, muy oportuno si el libro sufrió cambios directos en el contenido que no fueron incrustados en la estructura XML, como es muy común cuando la obra se maquetó para impresión.
- Habilitación de la opción `Dividir documento`. Esta obra tiene la intención de comenzar un nuevo apartado en cada

encabezado de primera jerarquía, así que la división será con un `h1` en `Estilo de párrafo único`. Podemos decir que la división es al EPUB lo que el salto de página es al PDF.

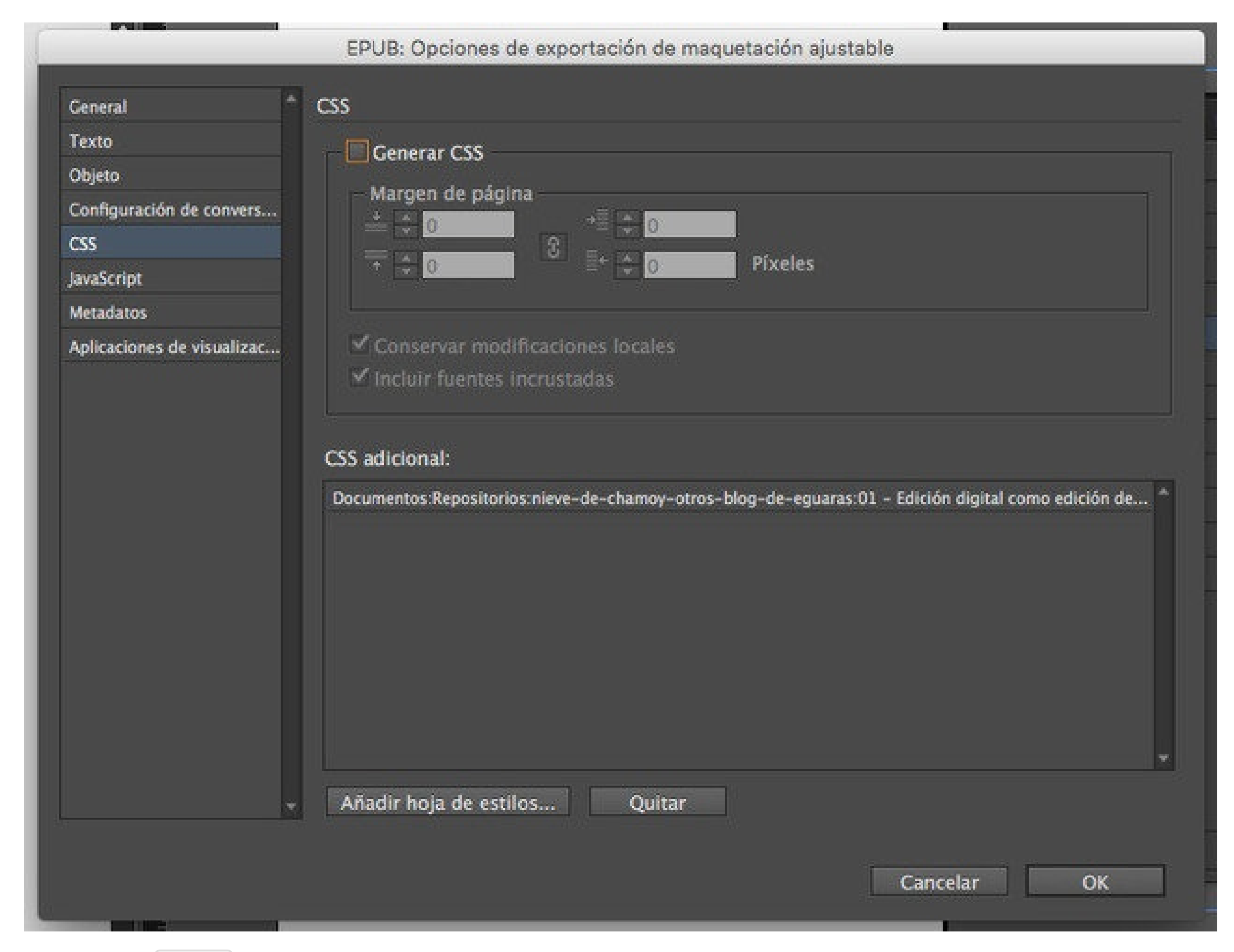


Configuración general en las opciones de exportación.

El segundo apartado es `CSS`. Si no se desea agregar una hoja de estilos CSS puede ignorarse lo siguiente.

- Para evitar que InDesign cree estilos CSS hay que desactivar la opción `Generar CSS`.

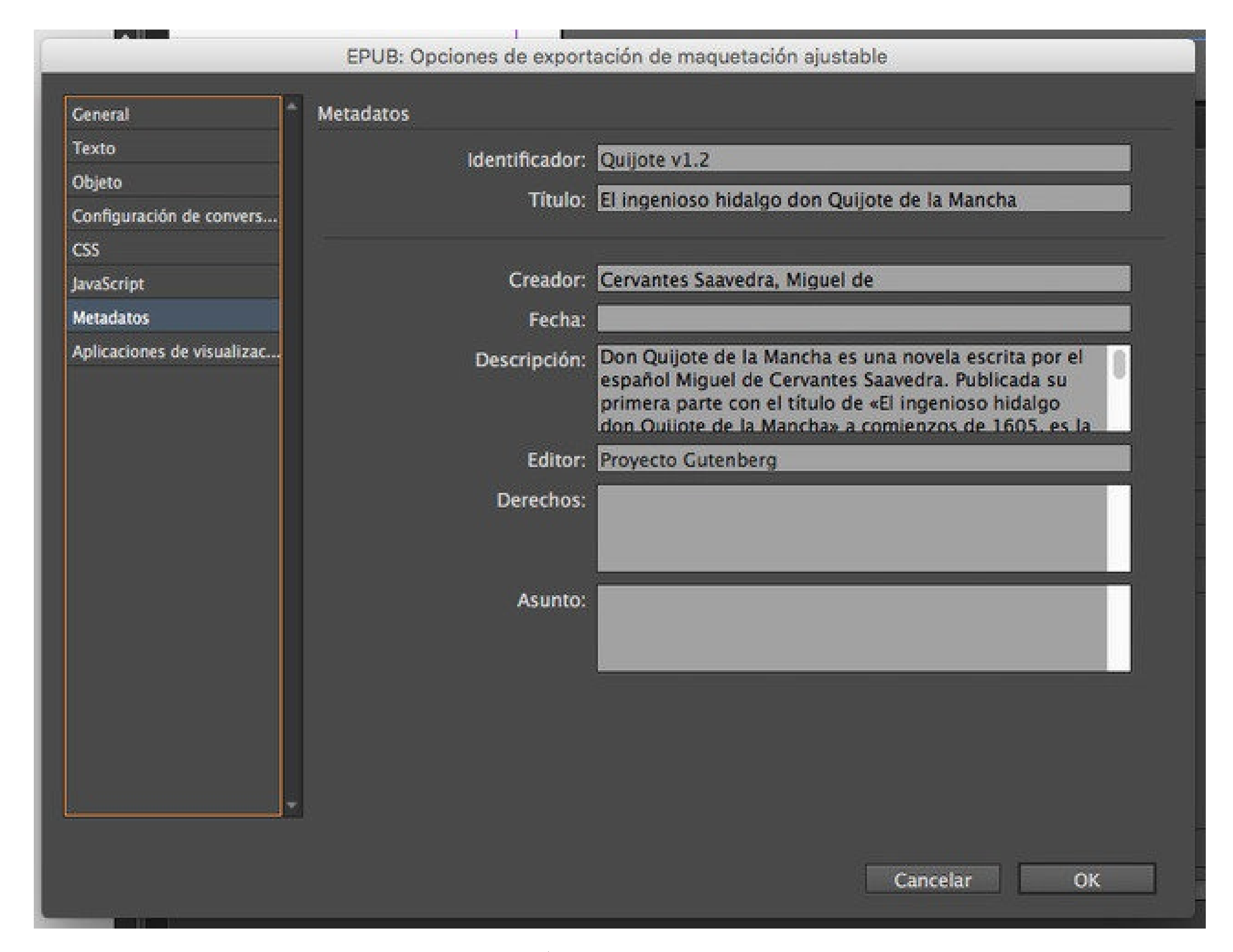
- Para agregar una hoja de estilos externa hay que hacer clic en `Añadir hoja de estilos…`.



Apartado `CSS` en las opciones de exportación.

Por último nos vamos al apartado `Metadatos`. Los **metadatos** son muy importantes para un *ebook* ya que, como dice Mariana, son «imprescindibles para la identificación de un libro». Cabe decir que **los metadatos son al EPUB lo que la ficha catalográfica es al impreso**. De esta manera, nos será muy fácil saber si cierto archivo es tal obra y no otra, sea en alguna tienda o en una biblioteca digital.

- `Identificador`. Este dato no es visible al usuario y sirve como mecanismo de control para quien edita. Nosotros preferimos los identificadores complicados por lo que solo incluimos el nombre del libro y la versión.
- `Título`. Aquí va el nombre de la obra que será visible para el usuario.
- `Creador`. En este campo se coloca el nombre del autor. Por convención, se recomienda escribir antes el apellido y después el nombre, separados por una coma. Si existe más de un autor, nosotros hemos decidido separarlo con punto y coma.
- `Fecha`. Aquí se recomienda colocar la fecha de publicación de la presente edición en lugar de la fecha de la primera edición.
- `Descripción`. Se trata de la sinopsis del libro, visible en algunas bibliotecas digitales o tiendas.
- `Editor`. Se indica el responsable de la edición o editorial.
- `Derechos`. En este espacio se indica qué tipos de derechos de autor tiene la obra.
- `Asunto`. Por último, mencionamos la categoría del libro. Nosotros separamos las categorías por comas: por ejemplo, «Ficción, Novela de caballerías». Si se desea una categorización más estandarizada, pueden utilizarse códigos BISAC tal como es necesario indicarlos si el libro se sube a iBooks, Google Play Books o Amazon.



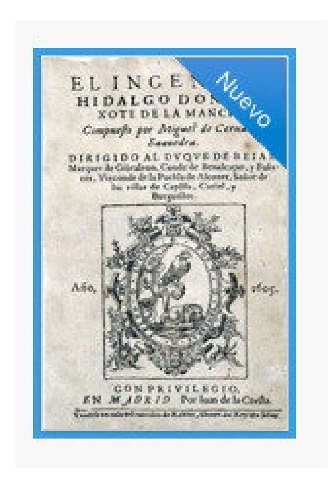
Metadatos en las opciones de exportación.

Ahora sí ya hemos configurado el EPUB por lo que solo resta **dar clic en OK ¡para generar el EPUB!**

6. El EPUB resultante

Nuestro EPUB está listo. Este tiene:

- una portada visible en la estantería;



Thumbnail del EPUB.

- un índice, y

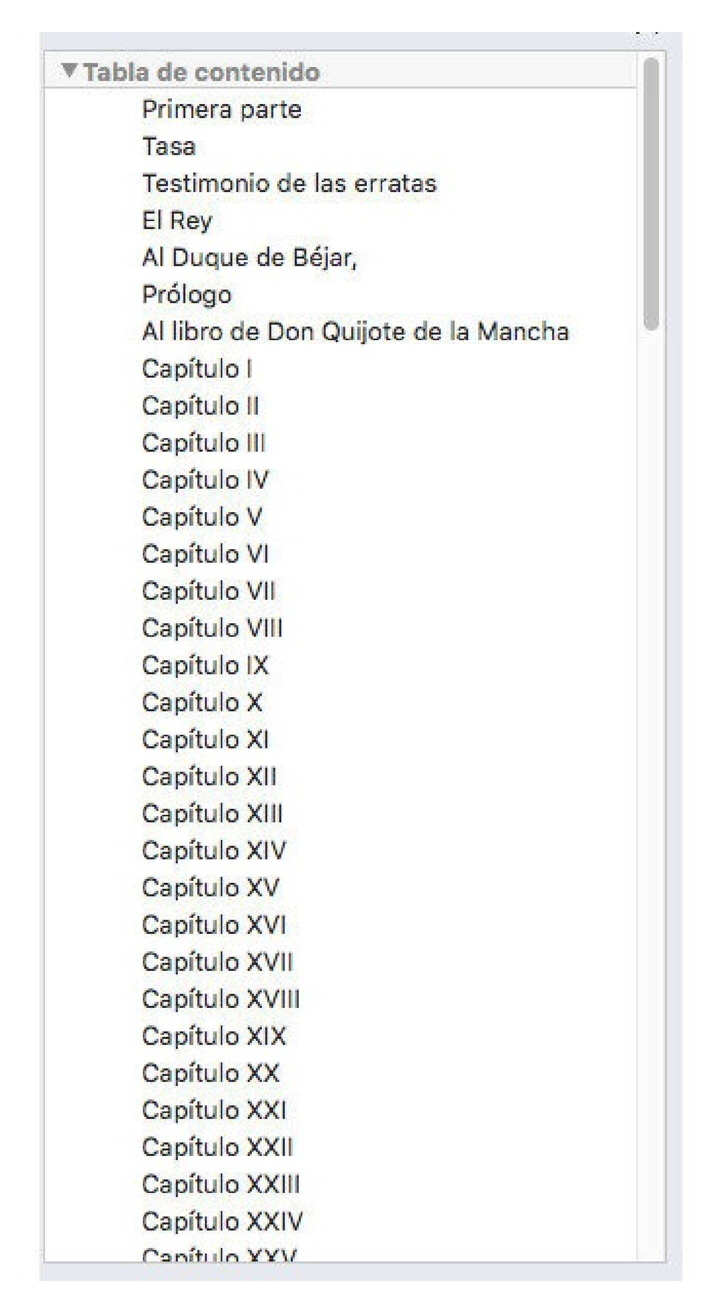


Tabla de contenidos del EPUB, también llamado índice en español.

- un contenido controlado a través de una estructura XML y una hoja de estilos CSS.

Capítulo I

Que trata de la condición y ejercicio del famoso hidalgo don Quijote de la Mancha

En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo que vivía un hidalgo de los de lanza en astillero, adarga antigua, rocín flaco y galgo corredor. Una olla de algo más vaca que carnero, salpicón las más noches, duelos y quebrantos los sábados, lantejas los viernes, algún palomino de añadidura los domingos, consumían las tres partes de su hacienda. El resto della concluían sayo de velarte, calzas de velludo para las fiestas, con sus pantuflos de lo mesmo, y los días de entresemana se honraba con su vellorí de lo más fino. Tenía en su casa una ama que pasaba de los cuarenta, y una sobrina que no llegaba a los veinte, y un mozo de campo y plaza, que así ensillaba el rocín como tomaba la podadera. Frisaba la edad de nuestro hidalgo con los cincuenta años; era de complexión recia, seco de carnes, enjuto de rostro, gran madrugador y amigo de la caza. Quieren decir que tenía el sobrenombre de Quijada, o Quesada, que en esto hay alguna diferencia en los autores que deste caso escriben; aunque, por conjeturas verosímiles, se deja entender que se llamaba Quejana. Pero esto importa poco a nuestro cuento; basta que en la narración dél no se salga un punto de la verdad.

Es, pues, de saber que este sobredicho hidalgo, los ratos que estaba ocioso, que eran los más del año, se daba a leer libros de caballerías, con

Contenido del EPUB.

Para acceder a todos los documentos empleados para esta edición, puedes descargar este archivo. Al descomprimirse encontrarás que también están presentes los archivos empleados para los otros dos métodos que quedan por describir: con Sigil y «desde cero». Para los archivos usados en la versión de InDesign solo hay que ir a `epubs > 1-indesign-cc`.

Conclusiones

Para el usuario de InDesign que está acostumbrado a trabajar con estilos directos o con estilos de párrafo y caracteres esta descripción puede parecerle compleja, ya que involucra la elaboración de archivos de manera externa al flujo común de este *software*. Como puede observarse, estos documentos externos requieren algún conocimiento básico de HTML y CSS. Por este motivo, se mencionó que se descansaría *un poco* del código.

El proceso aquí descrito está pensado para alcanzar el máximo control y la mayor calidad en un libro EPUB hecho con InDesign. Sin embargo, de manera irremediable implica vérselas con lenguajes de marcado. Es decir, incluso en el programa más común para la creación de libros es menester trabajar desde las entrañas de un EPUB.

Gracias a esto podemos indicar que:

1. para fortuna o desgracia el «código» llegó para quedarse en la edición digital si el objetivo es la creación de publicaciones de gran calidad;
2. InDesign no puede abarcar todo el flujo de trabajo como hace algunos años lo había hecho, ya que su arquitectura está pensada para la maquetación de libros en formato PDF (con los años se ha visto obligada a posibilitar la creación de *ebooks*);
3. es necesario un conjunto de herramientas o un *software* que nos permita atajar los retos de la publicación digital de una manera más directa y aún más controlada, y
4. por esta necesidad de diversidad en el ecosistema del libro se hace necesario optar por un proceso de producción que

simplifique y automatice el reto de publicar una obra en diversos formatos.

Debido a estas cuestiones y las desventajas que presenta un EPUB creado en InDesign (indicadas en «Edición digital como edición desde cero)» es posible concluir que **InDesign nos permite crear libros EPUB de gran calidad, pero a través de una metodología que puede complicar aún más el proceso de producción** o que potencialmente implica una pérdida de control tanto en la estructura como en la edición y el diseño.

¿Cuál es el volumen de producción a manejar? ¿Qué nivel de control se desea sobre la edición? ¿Cuál es la disponibilidad para analizar y probar otras metodologías de producción? Son solo algunas de las preguntas por las cuales podríamos saber si crear EPUB con InDesign es el método que necesitamos para el desarrollo de *ebooks*.

Sobre esta entrada

- Título original: «De XML a EPUB con InDesign».
- Título en el *blog* de Mariana: «Cómo crear un libro digital: de XML a EPUB con InDesign».
- Fecha de publicación: 8 de noviembre del 2016.
- Redacción original.
- Revisión de Mariana.

Cómo crear un libro digital: de XHTML a EPUB con Sigil

En esta entrada se detallan los pasos para hacer un libro electrónico. Es decir, qué acciones deben hacerse para obtener un archivo EPUB desde un XHTML con el programa Sigil.

Sigil, ven a mí

De todas las herramientas conocidas que existen para crear EPUB, Sigil es la que siempre recomiendo. Esto se debe a los siguientes motivos:

- Es *software* libre, por lo que no solo es gratuito o todo su código está disponible, sino que también implica que no hay necesidad de pedir permiso para usarlo o modificarlo.
- Es la opción popular más ligera, su peso va de ~45 a \~60 MB, según el sistema operativo que utilices.
- Es un entorno gráfico pensado únicamente para crear EPUB; es decir, tenemos la certeza que cuenta con las herramientas necesarias para hacer un buen EPUB de modo visual.

Sigil fue el único programa que satisfizo mis primeras necesidades para el desarrollo de EPUB. Uno de los principales motivos fue porque en aquel entonces andaba sobre Ubuntu, por lo que me era imposible usar *software* creado para Windows o Mac.

Pero otra razón fue que mis conocimientos en HTML y CSS eran

muy básicos y tampoco tenía un dominio sobre la terminal o RegEx. Muchos de ustedes se encuentran en esta situación, así que si quieren experimentar en la creación de EPUB, en lugar de InDesign —el cual es menos controlable, como ya lo indiqué—, los invito a que prueben con Sigil.

¿Cómo empezamos? Una ***primera desventaja*** de Sigil es que la documentación existente para su uso no es tan amigable como la de otras herramientas populares como InDesign o Jutoh. Cuando conocí Sigil, su instalación me provocó un dolor de cabeza. Por fortuna, hoy es fácil instalarlo en cualquier computadora.

El área de su documentación ha estado mejorando con el tiempo y aunque hace muchos años ya no uso Sigil, quizá este tutorial podría ser un buen punto de partida. Después de ahí, mi sugerencia es que metan sus manos al lodo y con los errores vean las posibilidades y límites de Sigil.

Ejemplo de EPUB

Como ya venía anticipado en mi primera colaboración con Mariana, vamos a tomar la edición del Proyecto Gutenberg del *Don Quijote* junto con algunas modificaciones en su estructura para producir un EPUB más limpio y con mejor diseño. Los archivos de ejemplo pueden descargarse aquí.

Con estos archivos nos será posible obviar el más grande problema en la edición digital: el formato. Como siempre he indicado, el cuello de botella en la edición, sea impresa o digital, es la falta de un formato adecuado. Esto provoca que gran parte del tiempo de desarrollo de una publicación sea, precisamente, la limpieza del formato. Vuelvo a insistir en el asunto solo para explicitar que:

1. este breve tutorial pasa por alto el tiempo necesario para dar un formato adecuado a una publicación y que muchos de ustedes, cuando empiecen a experimentar, se darán cuenta de la monotonía y frustración que implica trabajar de manera gráfica, y
2. esto implica que no nos enfocaremos en los aspectos básicos de la creación del EPUB, debido a que ya está cubierto en los archivos de ejemplo.

Orientaré esta entrada a ciertos errores comunes que he encontrado en los libros hechos por otros con Sigil, y que será una buena guía para que ustedes puedan evitarlos.

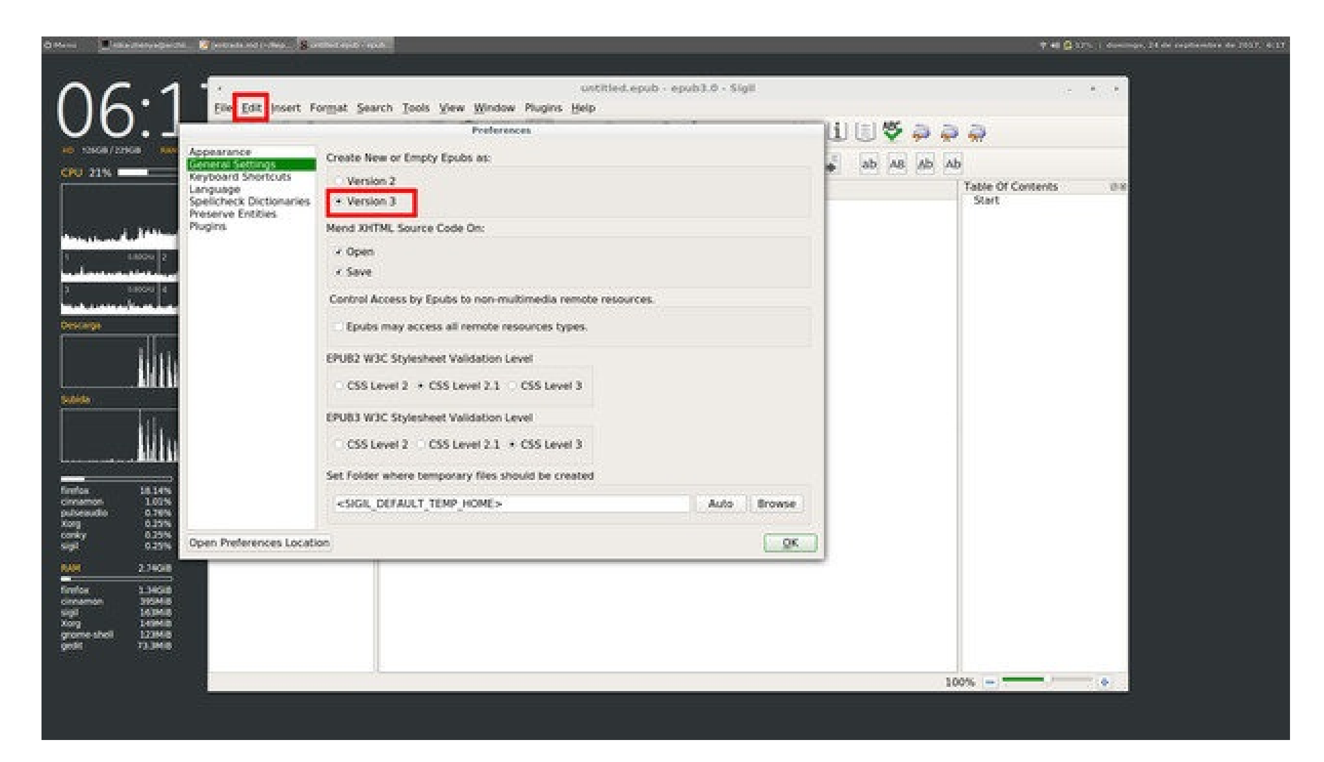
Manos a la obra

1. Mínima configuración general

Uno de los errores más comunes es olvidar la configuración general de Sigil. Existen dos casos donde esto afecta directamente al EPUB:

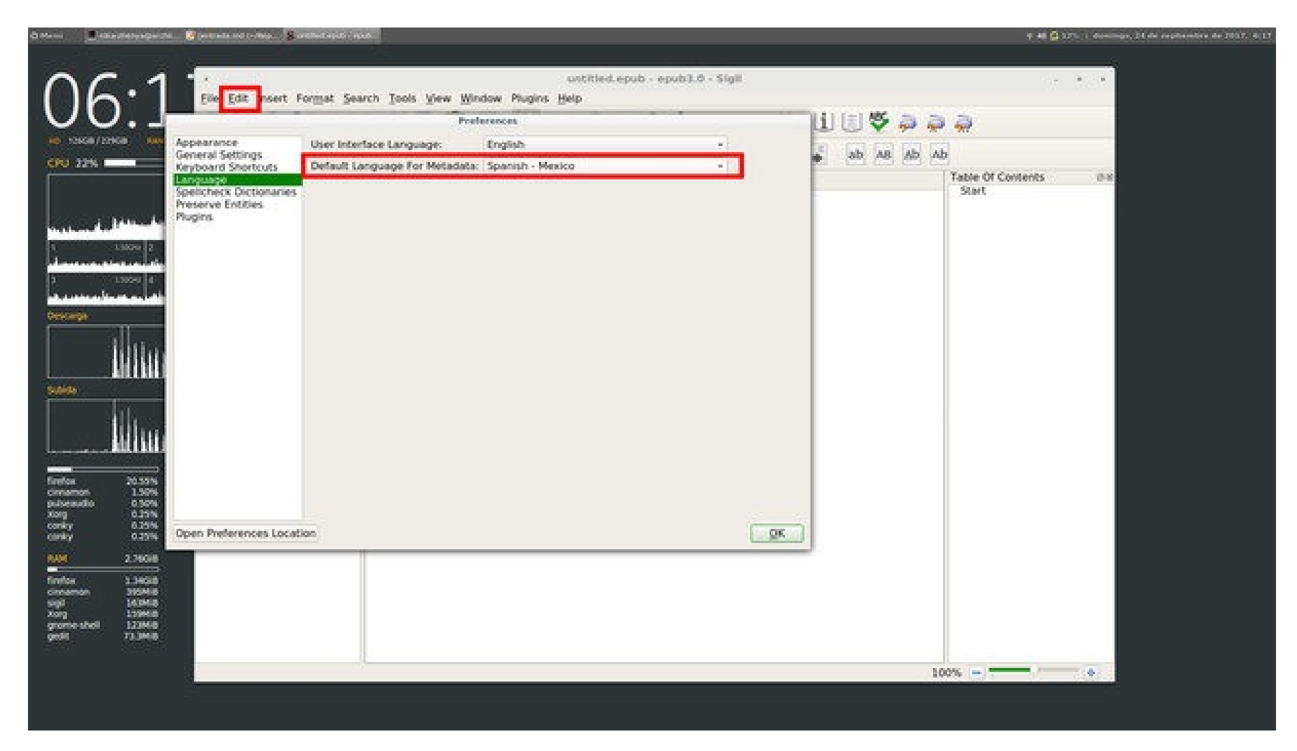
1. por defecto Sigil crea EPUB versión 2.0.1 cuando por lo menos es necesario crear a partir de la versión 3.0.0;
2. usa el idioma inglés como lenguaje por defecto para los metadatos del *ebook*.

Para cambiar esto es necesario ir a `Edit > Preferences…` o presionar `F5`. En el área de `General Settings` indicamos que deseamos la versión 3.



Cambio de versión a la 3.0.0.

Para especificar el lenguaje deseado de nuestros metadatos en la misma ventana vamos a `Language` y en `Default Language For Metadata` seleccionamos el idioma y la región de nuestra preferencia.

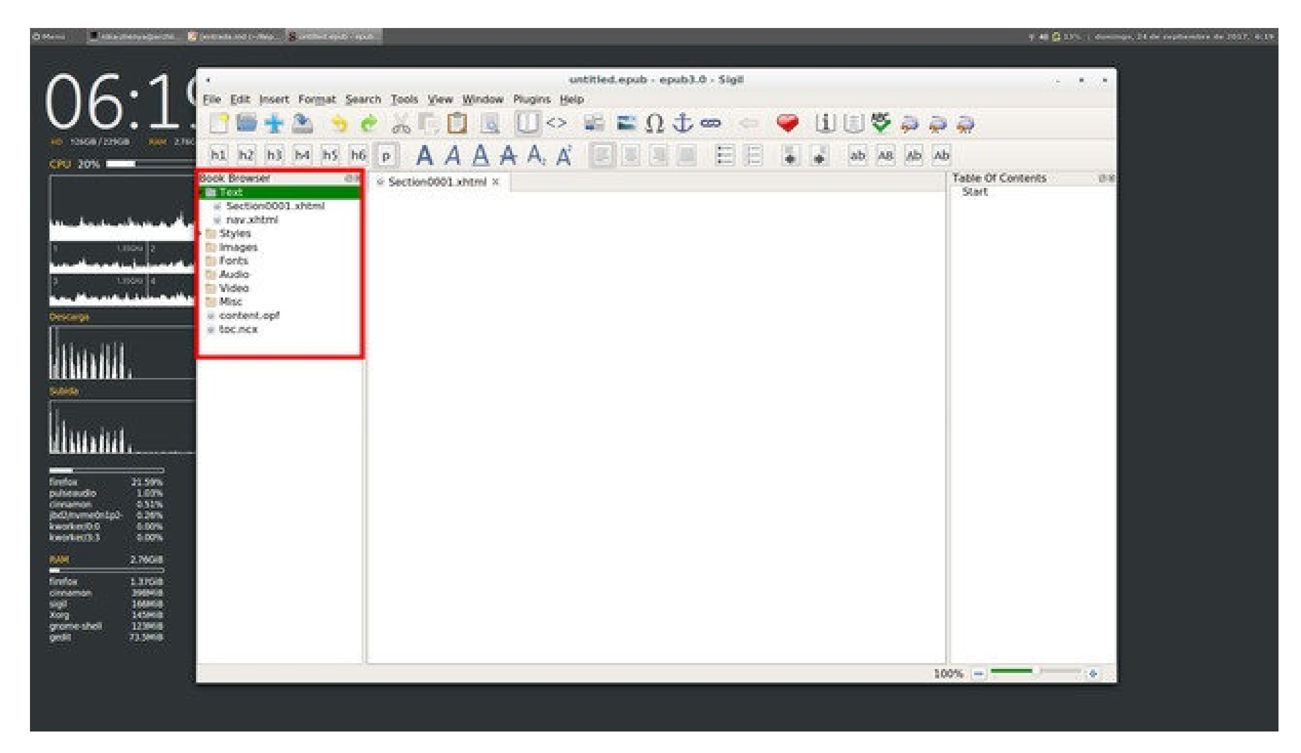


Cambio de idioma.

Para que los cambios surtan efecto lo recomendable es reiniciar Sigil. Con esto ya podremos crear EPUB versión 3.0.0 que, si bien es la versión más popular y aceptada por distribuidores —p. ej. Amazon, iTunes o Google—, no es la más reciente. Una ***segunda desventaja*** de Sigil es que aún no genera EPUB versión 3.1 o al menos 3.0.1.

2. Importación de archivos

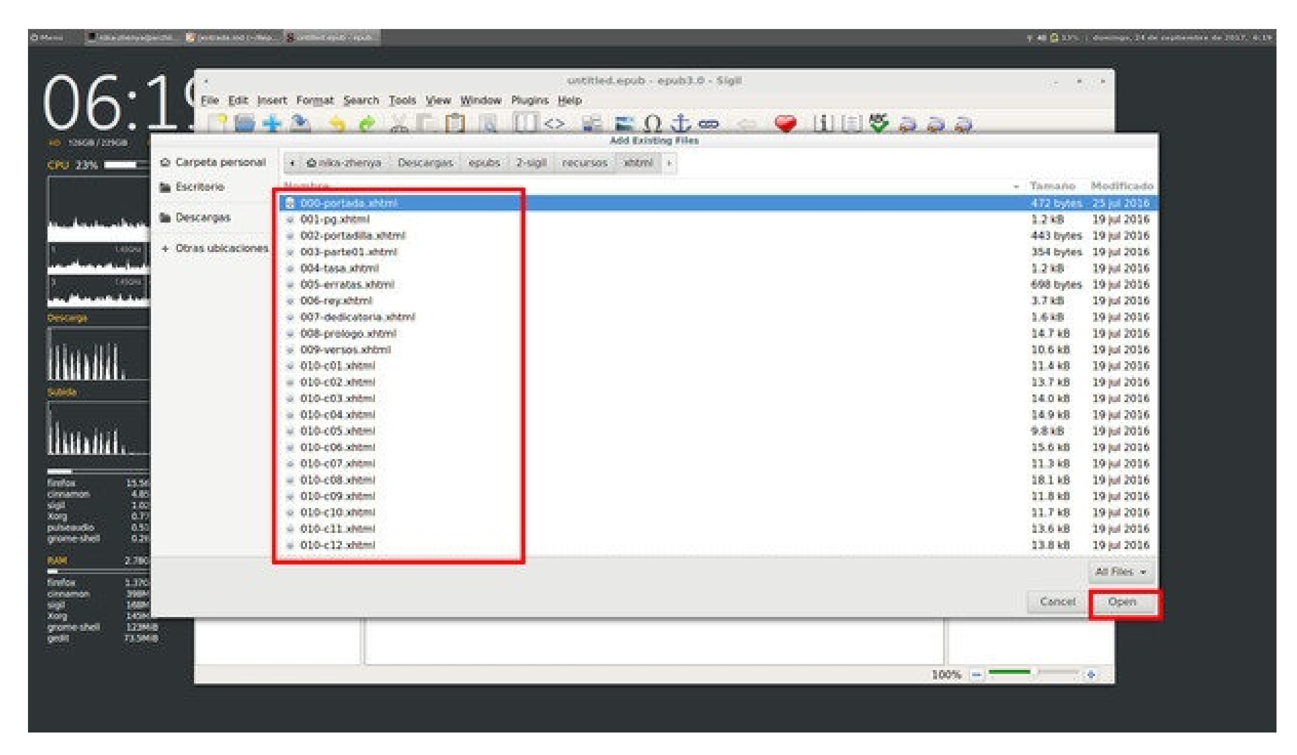
Cuando abrimos Sigil o generamos un nuevo proyecto veremos la siguiente estructura:



Estructura básica generada por Sigil.

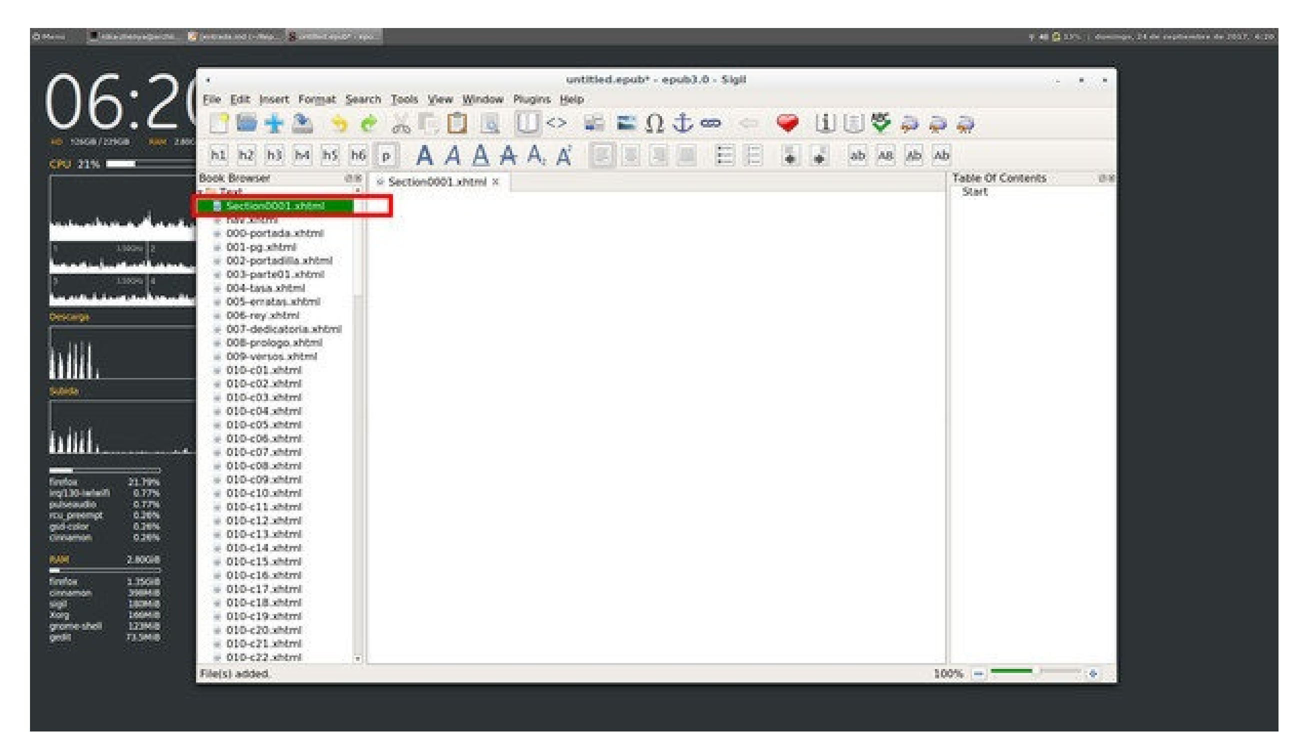
Como puede observarse, tenemos una serie de carpetas para poner ahí nuestros contenidos, como texto, hojas de estilo, imágenes, fuentes, audios, videos u otros elementos.

Para importar los XHTML de nuestros archivos de ejemplo hay que seleccionar la carpeta `Text` y con clic derecho escoger `Add Existing Files…`. Esto nos abrirá esta ventana:



Ventana de importación.

De nuestros archivos ejemplo vamos a seleccionar todos los XHTML presentes en `epub > 2-sigil > recursos > xhtml`. Así es como estos documentos serán incrustados en la carpeta donde van los textos del EPUB.

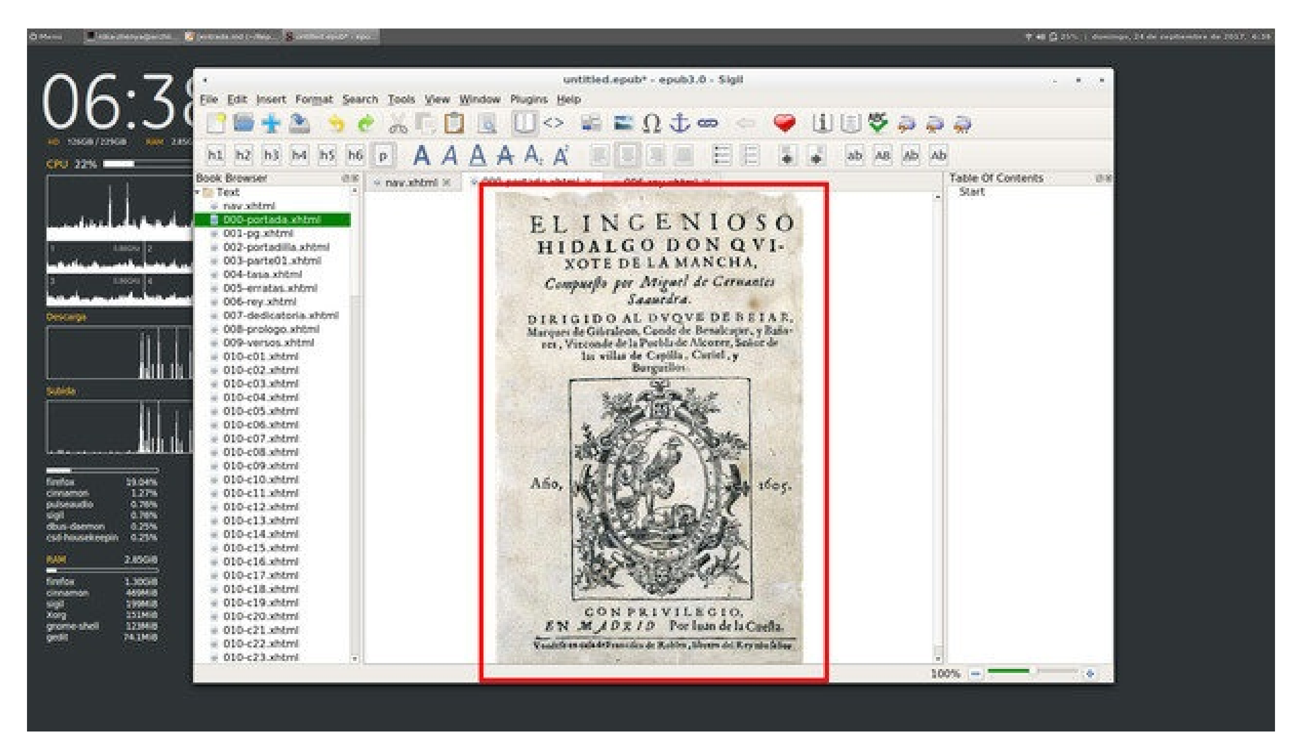


Archivos importadas y archivo innecesario.

El archivo `Section001.xhtml` no lo necesitamos así que lo seleccionamos y presionamos `Del` o clic derecho `Delete…` para eliminarlo.

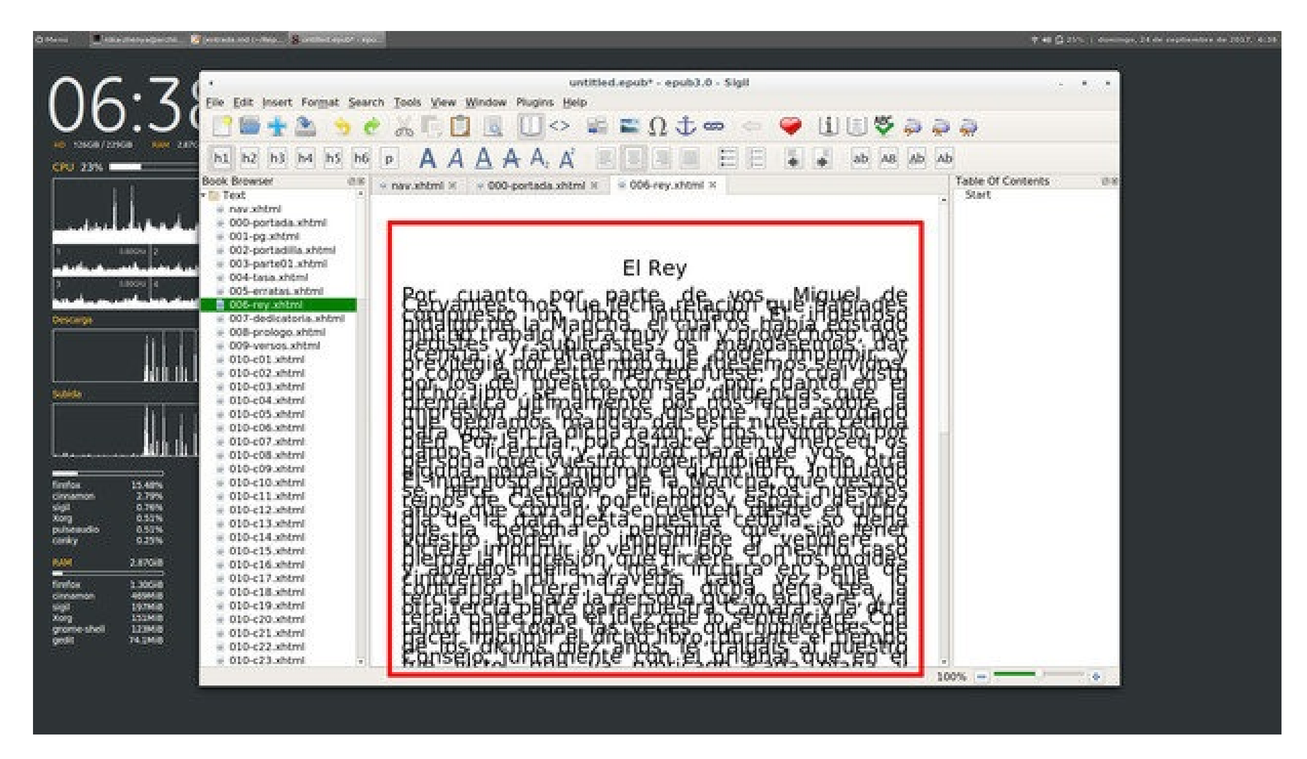
> *Recomendación*. Cuando generamos documentos XHTML desde Sigil, por defecto sigue la nomenclatura `Section` más un número consecutivo. Esto puede generar problemas al momento de verificar o hacer cambios ya que es un nombre genérico. Lo ideal sería darle nombre significativos y con una numeración inicial. Para mayor información, consulta estas recomendaciones.

Ahora bien, aquí tenemos dos elementos muy curiosos. El primero es que si damos doble clic al archivo `000-portada.xhtml`, ¡la imagen de portada ya está ahí!



Portada importada automáticamente.

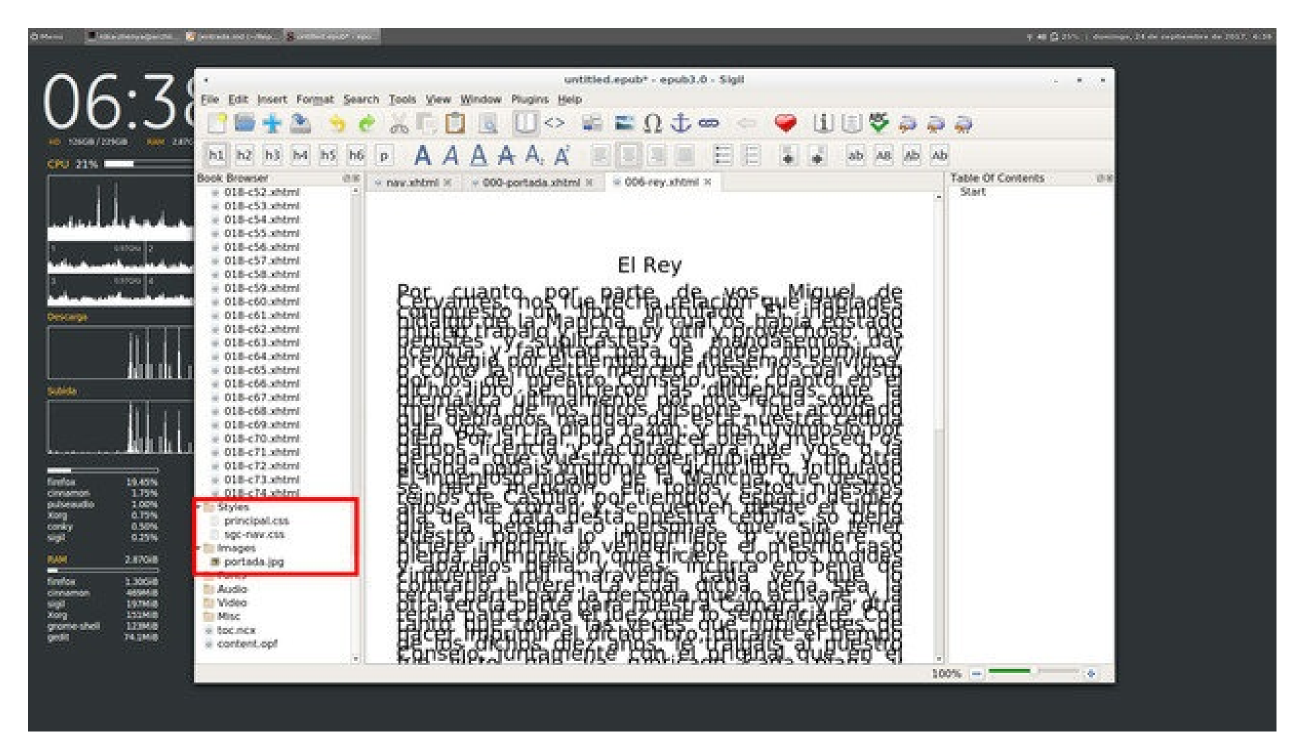
Sin embargo, si damos doble clic a `006-rey.xhtml` —u otro archivo que tenga texto—, vemos que los estilos son un desastre…



Estilos mal visualizados.

La ***tercera desventaja*** de Sigil es que en ocasiones no muestra el diseño correcto, por lo que una edición visual del contenido puede resultar muy problemática. Por fortuna, ahora sabemos que hay un error en la renderización de los estilos, así que vamos a obviar esta visualización —al final veremos cómo se visualiza realmente—.

¿Por qué se ven la portada y los estilos? Por defecto, Sigil importa todo aquello que esté vinculado a los archivos XHTML importados y los coloca en las carpetas correspondientes. En nuestro caso, `000-portada.xhtml` tiene vinculada la imagen `portada.jpg` que también fue importada a `Images`; además, todos los XHTML tienen un vínculo a la hoja de estilos `principal.css`, que fue colocada en la carpeta `Styles`.

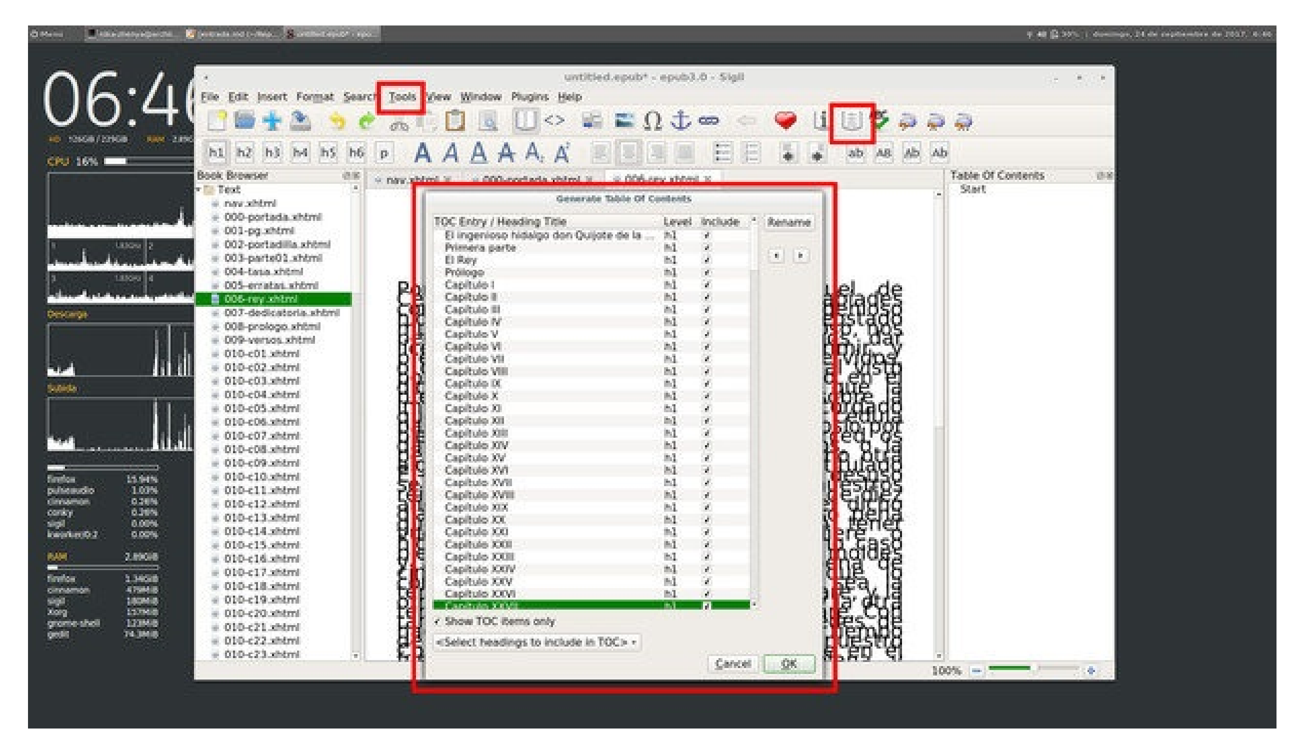


Archivos adicionales importados automáticamente.

3. Tabla de contenidos

Hasta aquí ya hemos terminado con los ajustes en nuestro contenido por lo que ya podríamos proceder a generar el EPUB. **Este es otro de los errores más comunes: el contenido y el diseño no lo son todo**. Existen otras tareas muy importantes que nos dan mejor control, mayor calidad y un producto más amigable para el lector.

Una de estas tareas es la modificación de la tabla de contenidos. Para acceder a esta vamos a `Tool > Table Of Contents > Generate Table Of Contents…` o presionando `Ctrl + T` o el icono resaltado.



Ventana de edición de la tabla de contenidos.

Sigil, de manera automática, detecta todos los encabezados en nuestro documento —etiquetas HTML de la `h1` a la `h6`—, podemos limitar su jerarquía en `<Select headings to include in TOC>`, eliminar elementos deseleccionando la caja `include` o crear una tabla anidada al usar las flechas para asignar jerarquías.

La ***cuarta desventaja*** de Sigil es que la creación de tablas anidadas también afectará la estructura de nuestro documento. Por ejemplo, esta obra consta de dos partes, así que lo ideal sería que cada una de las secciones interiores no fueran `h1` sino `h2` para que solo las partes queden en un nivel `h1`. Esta modificación haría que en cada uno de nuestros documentos también cambie el encabezado `h2` cuando, en realidad, lo único que queremos es *afectar la tabla de contenidos, no los contenidos*.

Por este problema, vamos a eliminar los elementos que se deseen pero también vamos a mantenerlos a todos con el nivel

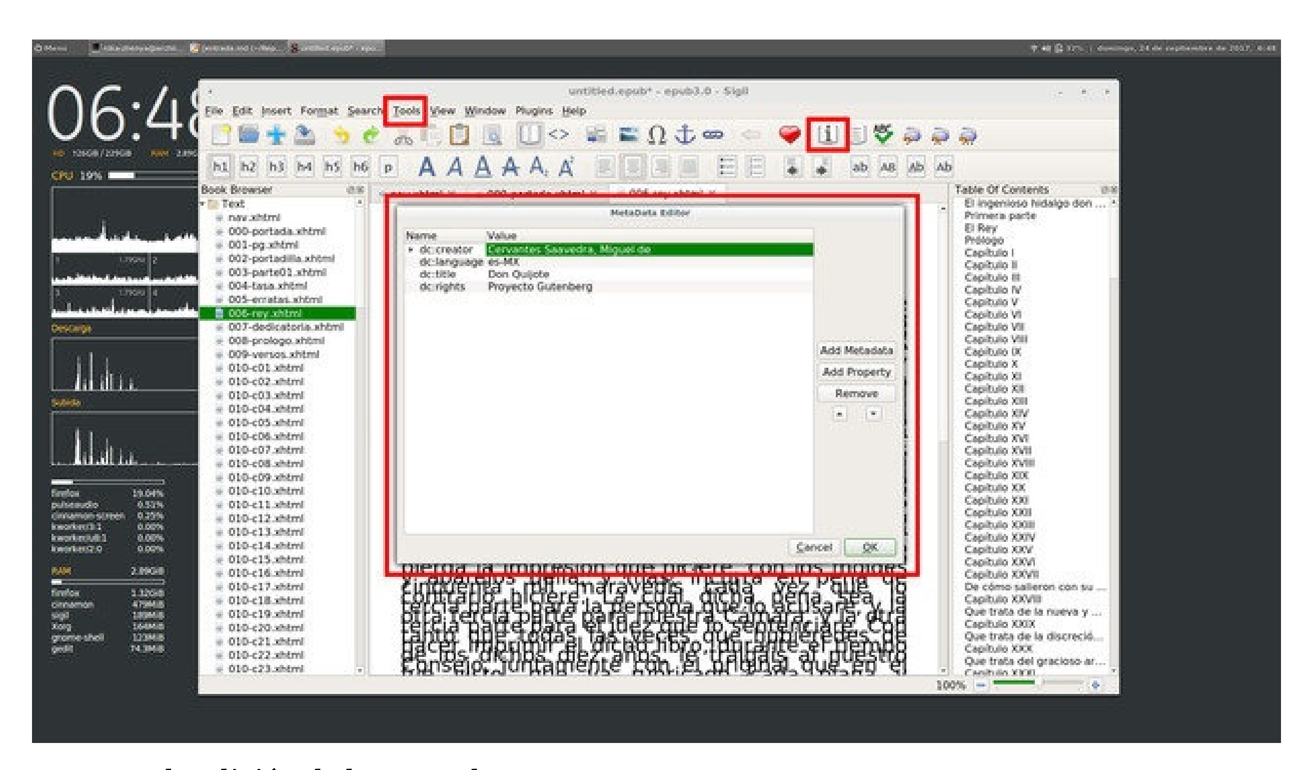
detectado. De lo contrario, tendremos una mala visualización en nuestro resultado final.

> *Aclaración*. Lo que en español conocemos por *índices* en el mundo anglófono se conoce como *table of contents*, y lo que nosotros denominados *índice analítico, índice onomástico*, etc., ellos lo llaman *indexes*. Para evitar confusiones, prefiero usar el término *tabla de contenidos*.

4. Metadatos

Los metadatos son uno de los elementos más importantes de un documento, como ya lo ha señalado Mariana. De manera general son el documento de identidad de nuestro libro, por lo que no pueden ignorarse y han de llenarse correctamente.

Para editar los metadatos nos vamos a `Tools > Metadata Editor…` o presionamos `F8` o el icono resaltado.



Ventana de edición de los metadatos.

Para añadir metadatos damos clic sobre `Add Metadata` y para modificar su valor hacemos doble clic sobre cada `value`. Para este ejemplo lo mínimo que requerimos es el autor, el lenguaje, el título y el titular de los derechos, pero ¿por qué no pruebas con agregar más campos?

> *Recomendación*. No existe consenso sobre la forma de escribir el nombre de los colaboradores; no obstante, es aconsejable usar la nomenclatura `Apellidos, Nombres`. Esto se debe a que en las bibliotecas digitales es posible ordenar los libros por nombre del autor.

En este punto, de nuevo podríamos generar nuestro EPUB, porque ya contamos con los elementos esenciales de un *ebook*:

- Contenido
- Diseño
- Tabla de contenidos
- Metadatos

Sin embargo, más vale limpiar un poco el proyecto para evitar potenciales errores en la verificación.

5. Regeneración y limpieza

Como hemos estado haciendo modificaciones a la tabla de contenidos y a los metadatos, **tenemos la necesidad de volver a generar los archivos** que guarden esta información. De no ser así, tengan por seguro que en la verificación aparecerán errores o advertencias.

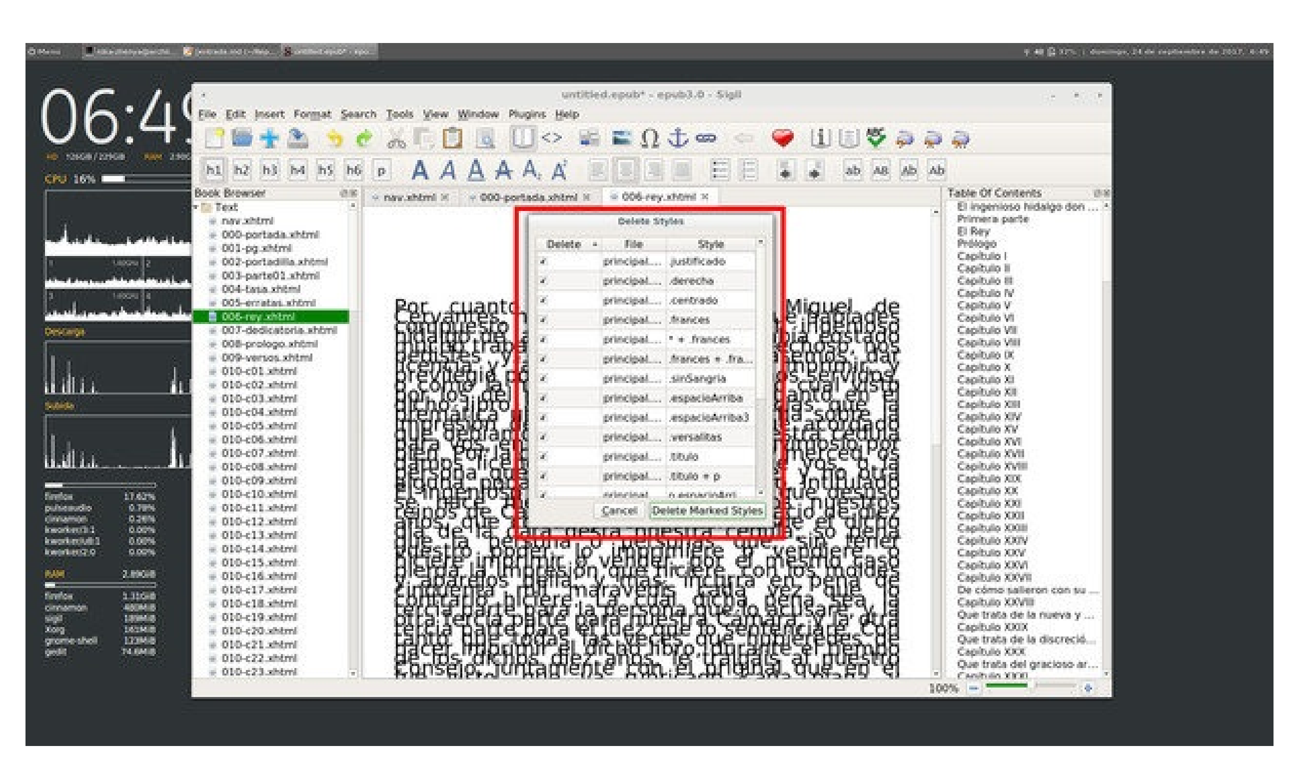
- Para regenerar el archivo con los metadatos vamos a `Tools > Epub3 Tools > Update Manifest`

`Properties`.
- Para regenerar las tablas de contenidos hacemos clic en `Tools > Epub3 Tools > Generate NCX From Nav`.

Aclaración. Los EPUB versión 3 tienen dos tablas de contenidos: un `nav.xhtml` para lectores recientes y un `toc.ncx` para dispositivos viejos.

Ahora procedemos a limpiar nuestro proyecto:

- Para eliminar archivos innecesarios hacemos clic en `Tools > Delete Unused Media Files…`.
- Para eliminar estilos que no se usan de las hojas de estilo, sino que vamos a `Tools > Delete Unused Stylesheet Classes…`.



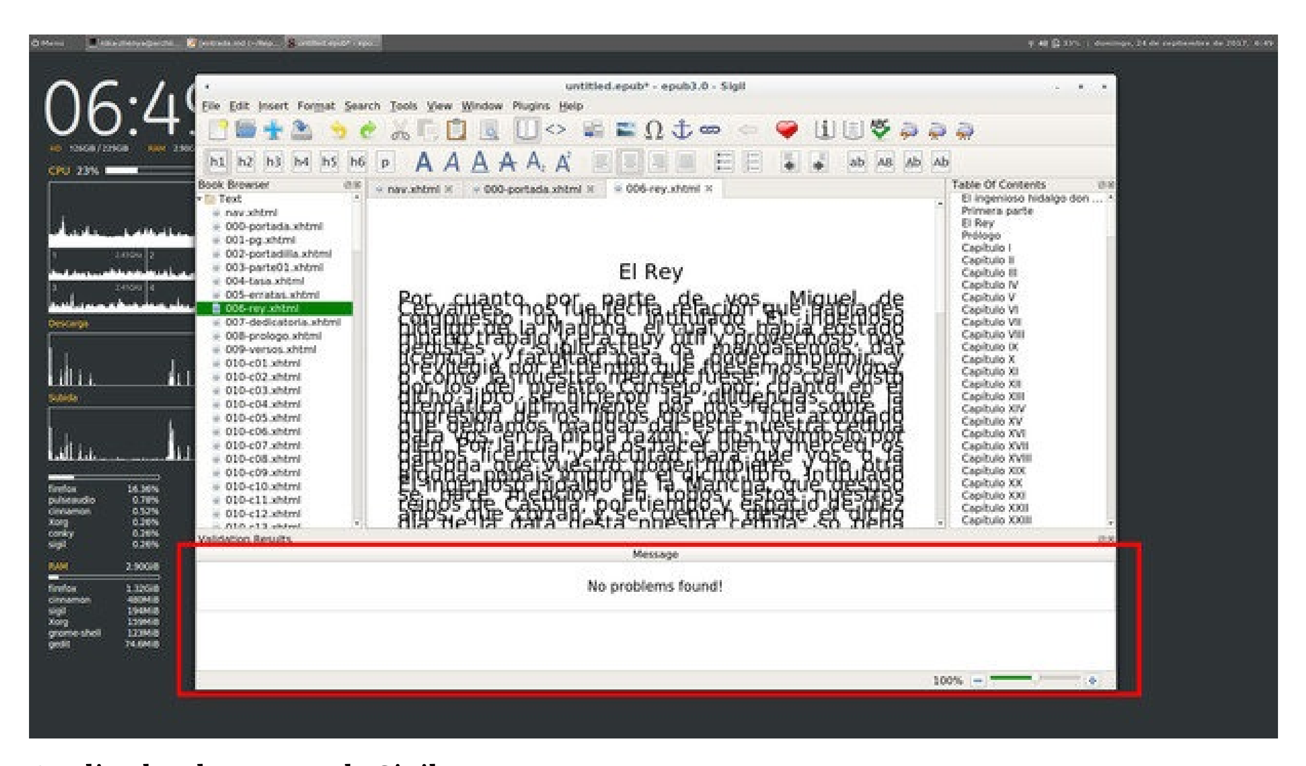
Ventana de eliminación de estilos no usados.

La hoja de estilos `principal.css` que está vinculada a los XHTML

es un archivo CSS con todas las clases que uso por defecto para cualquier clase de obra. Por este motivo, existen clases que no fueron necesarios para este libro, por lo que es detectado por Sigil y solo hacemos clic en `Delete Marked Styles`.

6. Creación y verificación

Sigil nos ofrece una herramienta para verificar el estado de nuestro EPUB, que está en `Tools > Well-Formed Check EPUB` o presionando `F7`.



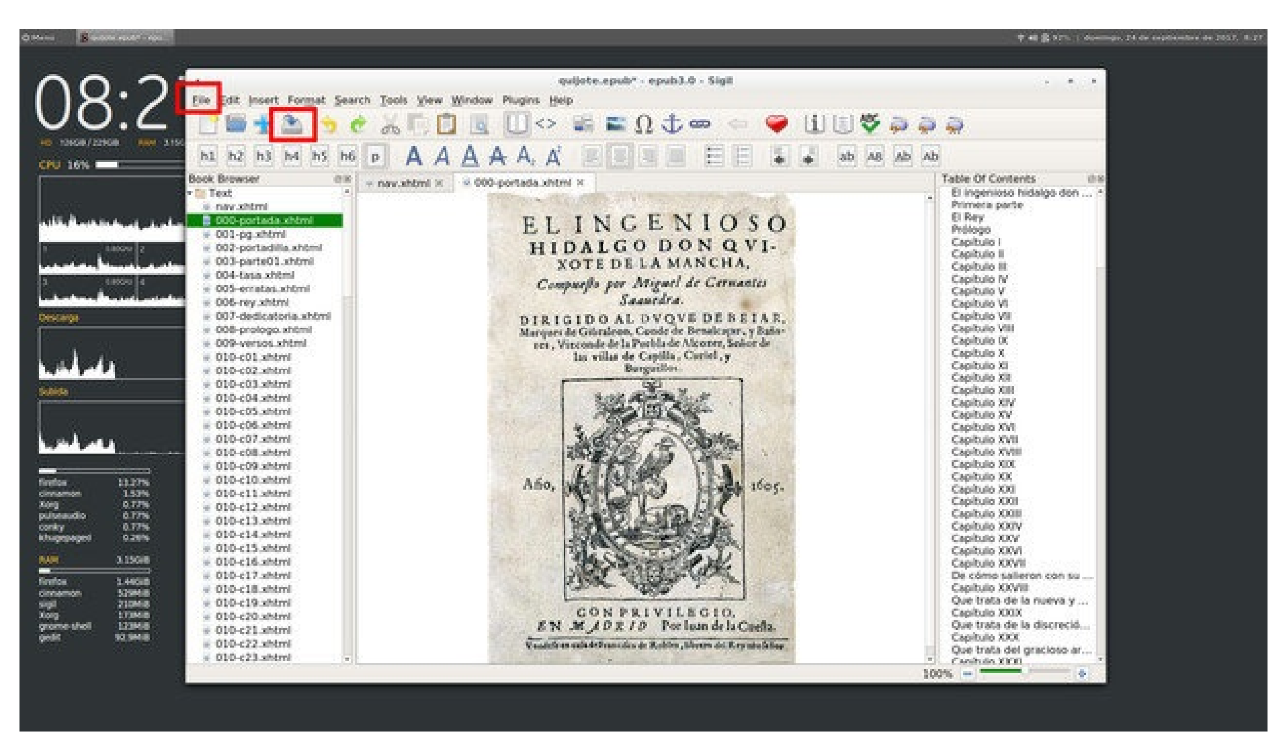
Analizador de errores de Sigil.

Nuestro proyecto no tendría que tener errores, pero evitemos engañarnos: esto no asegura que el resultado final esté exento de errores o advertencias. Esto me parece una ***quinta desventaja***, ya

que es común que el usuario piense que, por la ausencia de errores en este panel, no es necesario la verificación con herramientas externas.

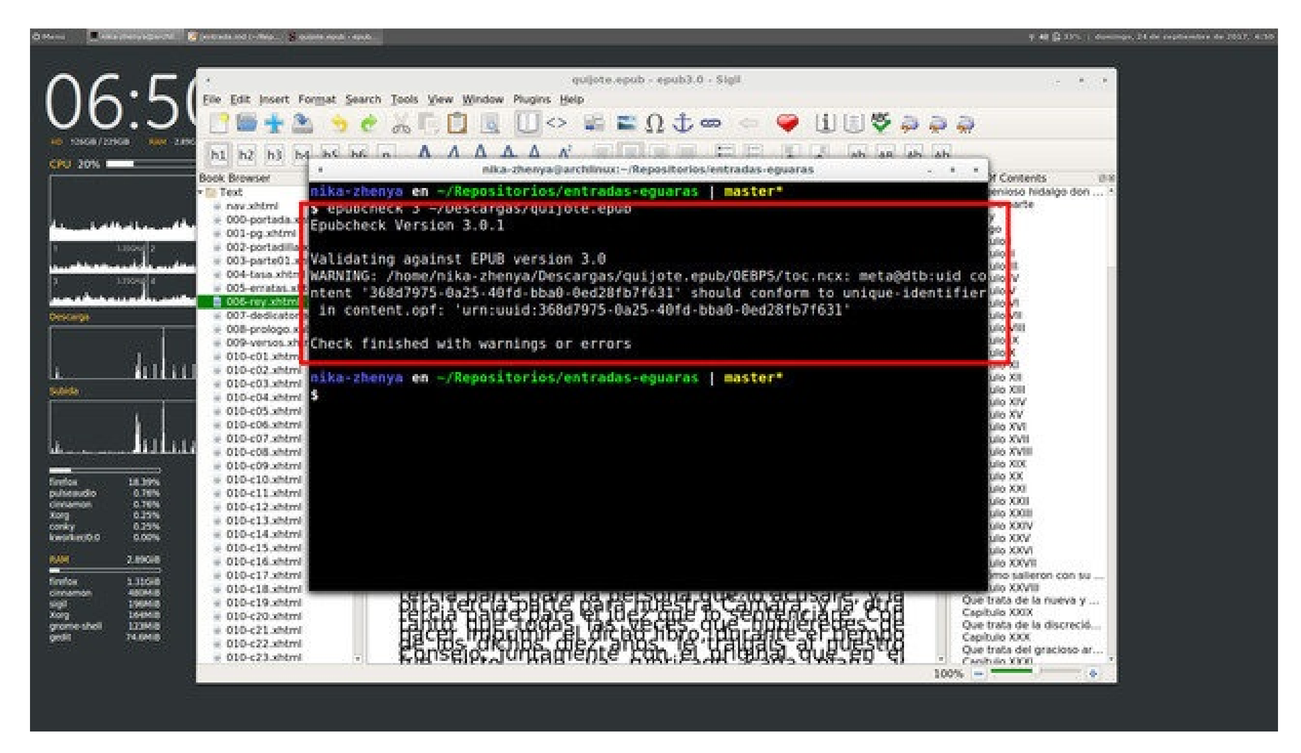
Esto es tan común que la mayoría de los *ebooks* que he descargado, por lo general, tienen errores o advertencias que en un determinado momento pueden afectar la experiencia de lectura.

Para demostrarlo, vamos a generar por primera vez nuestro libro desde `File > Save` o presionando `Ctrl + S` o el icono resaltado.



Generación de archivo EPUB.

Esto crea un archivo EPUB que se puede visualizar y, al parecer, no presenta problemas. Sin embargo, si usamos Epubcheck —la herramienta oficial para la verificación de EPUB desarrollada por el IDPF y que está disponible en línea o para su descarga— tenemos el siguiente resultado:



Primer resultado de Epubcheck.

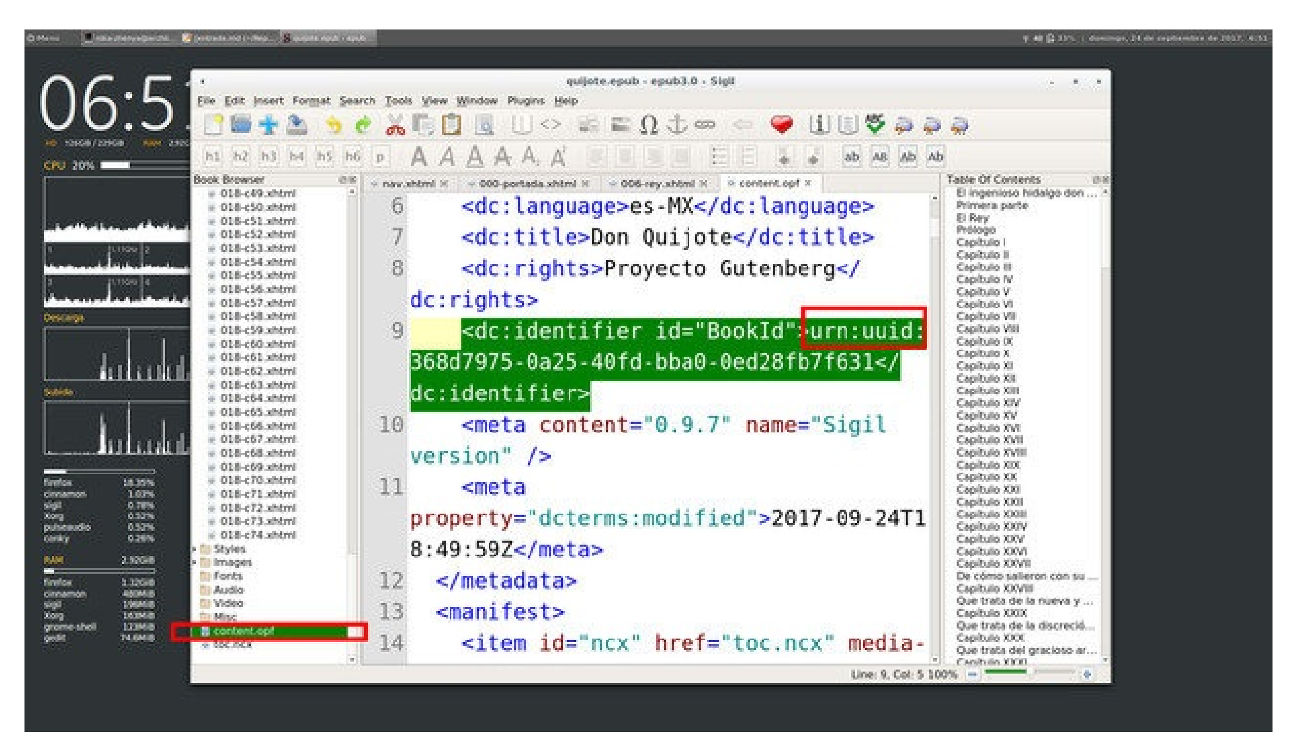
Se trata de una advertencia: no coincide el identificador presente en el archivo `toc.ncx` —el de las tablas de contenidos— con en el archivo `content.opf`, que contiene los metadatos, la declaración de contenidos y el orden de lectura del libro.

Siendo más específicos, en el `toc.ncx` tenemos un identificador de 32 caracteres separado por guiones. Mientras tanto, en el `content.opf` tenemos el mismo identificador pero con el sufijo `urn:uuid:`, por lo que hace que carezca de igualdad. Para evitar la advertencia solo tenemos que eliminar el sufijo. Sin embargo, estos vericuetos en la comprobación de un EPUB son fruto de la experiencia y uno que otro golpe al teclado a medianoche…

No lo considero una desventaja por el simple hecho de que ningún programa popular para la generación de EPUB cuenta con una herramienta externa de verificación o, mejor aún, una especie de depurador que nos asista durante la verificación del EPUB. Sin

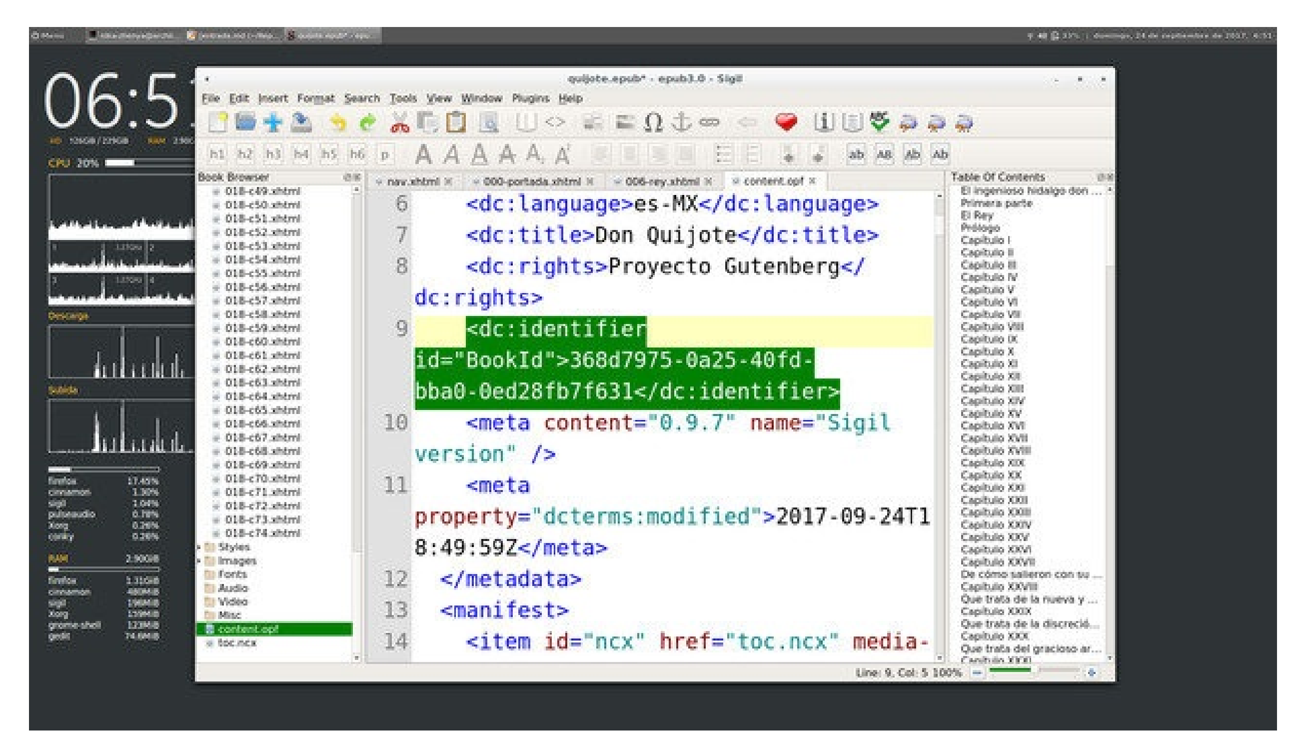
embargo, esto implica que **para la generación de *ebooks* sin ningún detalle técnico es menester el conocimiento de la estructura del EPUB**, que va más allá de lo visual.

Entonces, en Sigil vamos al archivo `content.opf` y eliminamos el sufijo.



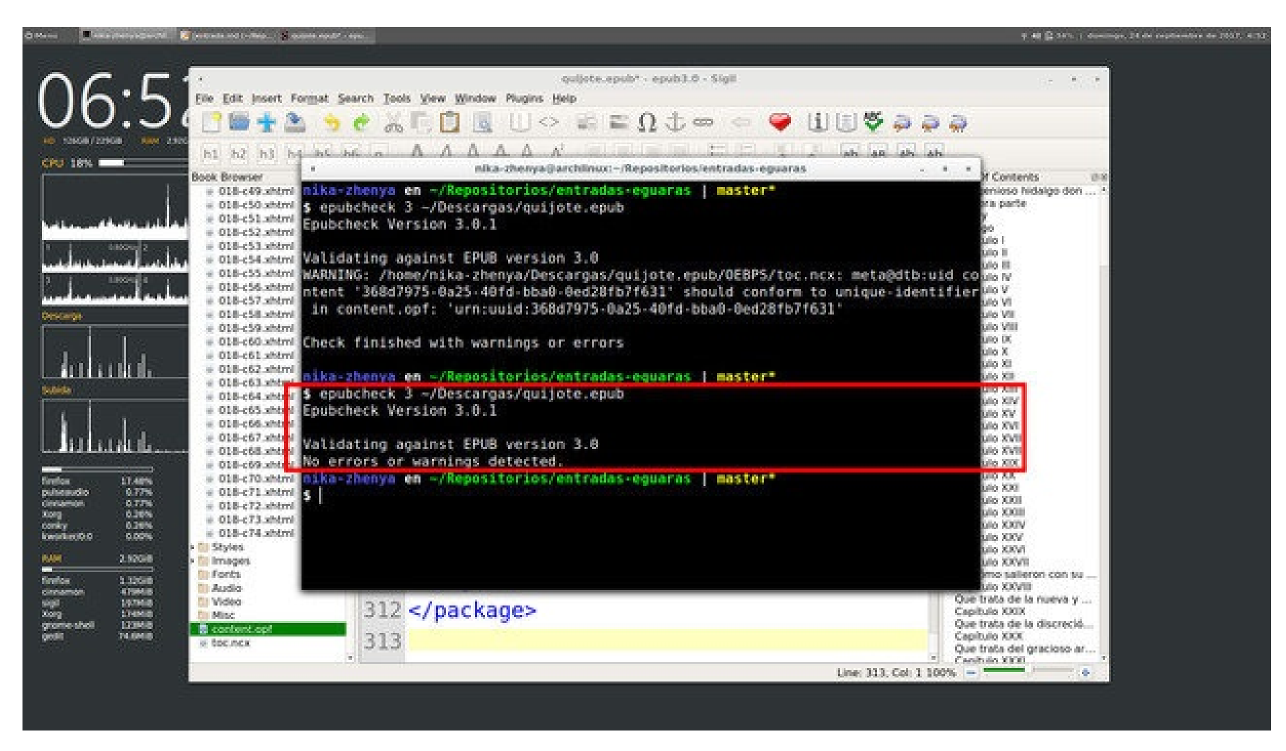
Localización del elemento a eliminar.

Una vez hecho esto, tendremos un resultado como el siguiente:



Archivo enmendado.

Ahora solo basta con volver a generar y verificar el EPUB para encontrar que ¡ya tenemos un EPUB sin errores ni advertencias!

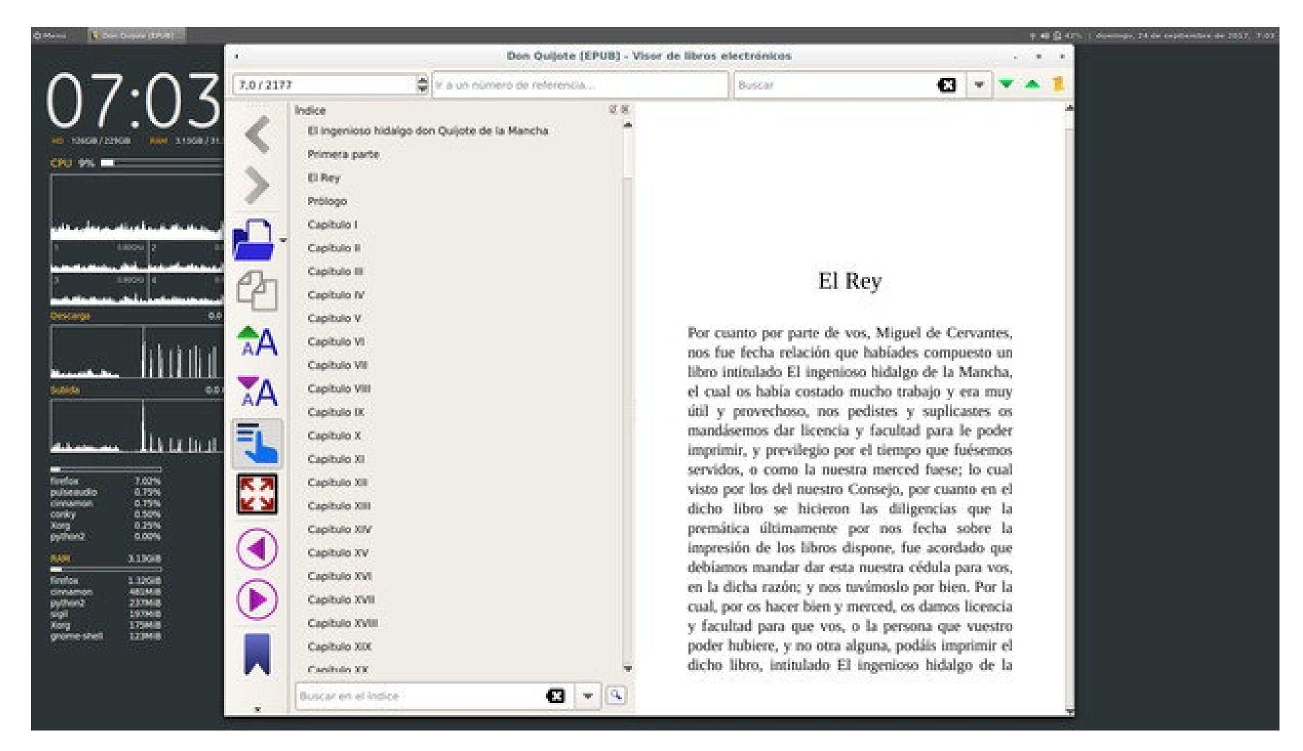


Verificación exitosa del EPUB.

> *Recomendación*. La descarga de Epubcheck implica usar la terminal, lo que en un primer momento es más difícil de usar. Sin embargo, presenta estas ventajas en comparación con su uso en línea: menor tiempo de verificación, carencia de límite en el peso del EPUB y posibilidad de empleo *offline*. En la medida posible, es mejor idea empezar a usar la versión descargable.

7. ¡Listo!

Ahora ya podemos abrir nuestro EPUB para verlo y, como puede observarse, tenemos la tabla de contenidos como la estipulamos y ¡los estilos se ven correctamente!



Visualización final del EPUB.

Este archivo podemos enviarlo a nuestro dispositivo favorito de lectura o subirlo a alguna tienda. Ahora bien, ¿por qué no intentas hacer un EPUB con otra obra?

> *Aclaración*. Por defecto, Sigil no modifica el identificador de libro en cada nueva generación que hacemos, además de que conserva el mismo nombre de archivo. Esto provoca ciertos problemas en algunos lectores. En iBooks existe la posibilidad de que abra el archivo más viejo, ya que al detectar el mismo identificador, mostrará el EPUB más antiguo sin importar que este haya sido eliminado. En Adobe Digital Edition se ha detectado la misma situación, pero con el nombre del archivo. Si experimentas este problema, prueba con cambiar el identificador en el `content.opf y el toc.ncx`, así como cambiar manualmente el nombre al archivo.

Sigil, siempre te recordaré

Las cinco desventajas que indiqué no fueron los motivos por los que

decidí dejar de usar Sigil; al final, para la mayoría de los lectores, editores o desarrolladores carece de importancia. **Con el tiempo opté por la creación desde cero, porque de manera visual no obtuve el control deseado y la verificación se volvía muy escabrosa.**

Más que un abandono de esta gran herramienta, me vi obligado a buscar otra metodología de desarrollo. La corrección de errores y advertencias me permitió conocer a fondo la estructura del EPUB, tanto del lenguaje HTML y CSS para el contenido, estructura y diseño del libro, como del XML que se emplea para la declaración de las tablas de contenidos, los metadatos, el manifiesto, la espina y otras particularidades. Bajo este aprendizaje, el desarrollo visual de un EPUB me terminó por consumir más tiempo.

Este empoderamiento tecnológico me ha permitido estar desarrollando una serie de **herramientas para el quehacer editorial** que se llaman **Pecas**. El último ejemplo que haremos con el *Don Quijote* será usando esas herramientas.

Sobre esta entrada

- Título original: «De XHTML a EPUB con Sigil».
- Título en el *blog* de Mariana: «Cómo crear un libro digital: de XHTML a EPUB con Sigil».
- Fecha de publicación: 3 de octubre de 2017.
- Redacción original.
- Revisión de Mariana.

De XHTML a EPUB y MOBI con Pecas, ¡en solo un minuto y medio!

> En el este artículo se utiliza Pecas, un conjunto de herramientas editoriales que aún están en desarrollo. Existen diferencias sustanciales en su uso desde la primera vez que se mencionó como «edición desde cero». Y su uso, probablemente, también varíe en el futuro.

El objetivo de esta entrada es explicar cómo producir **dos archivos EPUB y un MOBI validados en un minuto y medio**.

El último Don Quijote

Desde que comencé a colaborar en este blog hemos hecho versiones de *Don Quijote* con distintas herramientas.

- Primero empezamos con un análisis general en el que se explicaba que el método «desde cero» es el más pertinente para un quehacer editorial controlado.
- Después explicamos cómo hacer un EPUB con InDesign.
- Por último, mostramos cómo crearlo a partir de Sigil.

Hoy es el turno del último modo de producción: **el método desde cero**. Pero ¿qué es eso de «desde cero»? Muy sencillo: meternos de lleno con el código de nuestro libro.

Esto no implica que tengamos que conocer la estructura del EPUB —aunque es recomendable—, ya que es posible usar una serie de *scripts* que nos permiten concentrarnos solo en el contenido del libro.

¿Qué es Pecas?

Los *scripts* que utilizo y desarrollo para hacer cualquier tipo de publicación son herramientas libres que están disponibles aquí. El conjunto de estos *scripts* se llama Pecas y la intención es crear herramientas editoriales bajo las siguientes ideas:

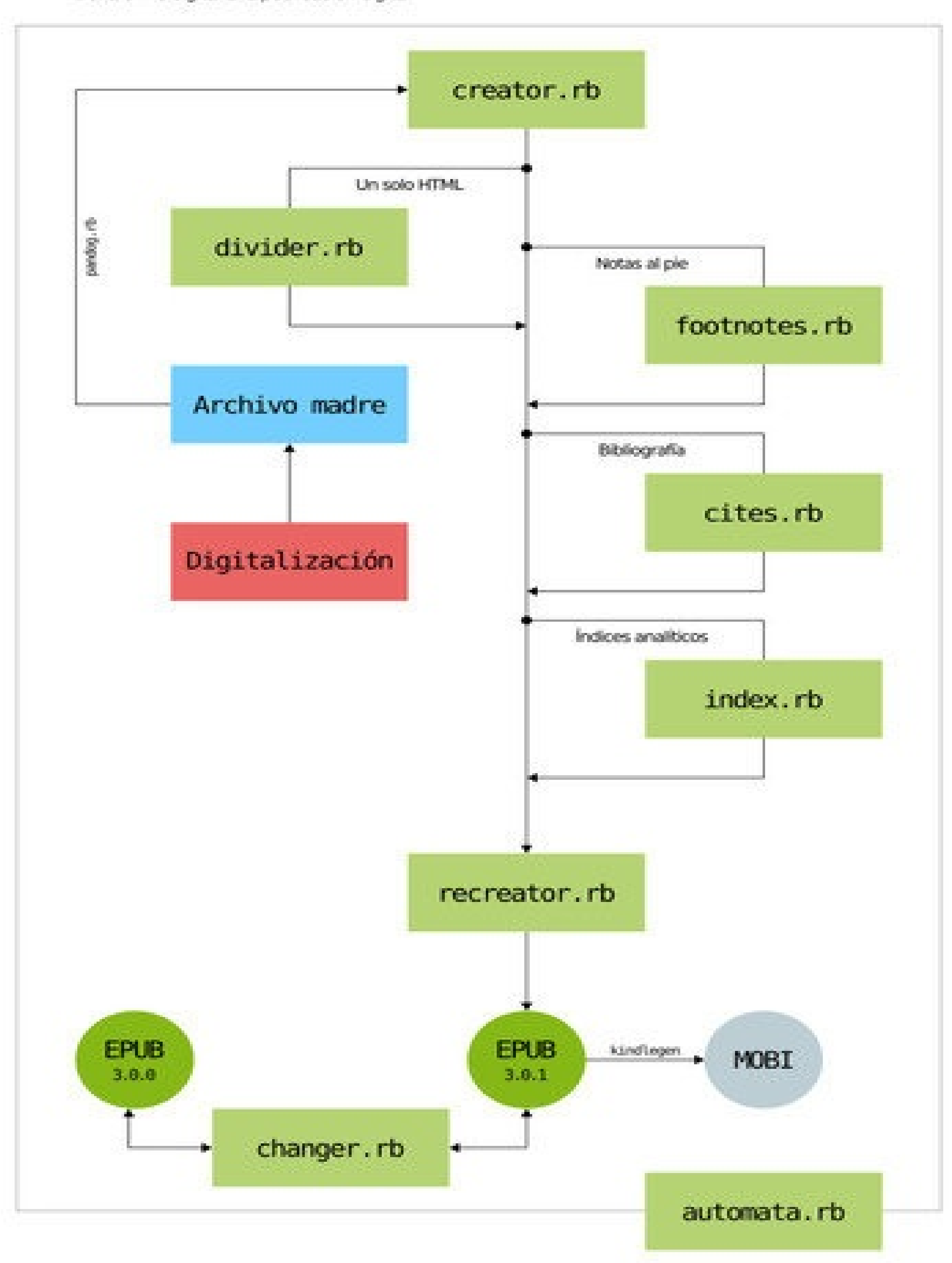
- **Desarrollo multiformato**, para obtener, al menos, archivos EPUB, MOBI y PDF para *ebook*, además de PDF para impresión a partir de archivos madre. Es lo que en otras publicaciones se ha llamado «edición ramificada».
- **Herramientas de control editorial** y no solo para la producción de formatos.
- **Formatos que se adapten** a la mayoría de las exigencias posibles de la tradición editorial en español.

Las herramientas por el momento carecen de entorno gráfico, por lo que se usan desde la terminal. Son multiplataforma, por lo que no importa si usas Windows, macOS o Linux (en Windows recomiendo instalar un WSL con Ubuntu para mayor comodidad).

Como puede notarse, es un proyecto muy ambicioso que ya lleva casi año y medio de vida. A estas alturas, el área de publicación digital está ya casi terminada y podemos verlo en este gráfico:

Proceso para EPUB

Del archivo digital a la publicación digital



Proceso para la producción de EPUB.

A partir de un archivo madre en Markdown o HTML es posible:

1. arrancar con la creación de un proyecto EPUB (`creator.rb`),
2. dividirlo en secciones (`divider.rb`),
3. añadirle notas al pie (`notes.rb`),
4. producir el EPUB (`recreator.rb`),
5. convertirlo a otras versiones de EPUB (`changer.rb`),
6. e incluso generar un MOBI si se cuenta con un programa desarrollado por Amazon (`kindlegen`).

Aún está pendiente el manejo de referencias bibliográficas (`cites.rb`) y la creación de índices analíticos o glosarios (`index.rb`).

Para los afortunados que no tienen manía por el control de cada uno de los procesos, existe la posibilidad de automatizarlos y hacer todo con un solo comando (`automata.rb`).

Con este libro procederemos de esta manera y **en un minuto y medio produciremos dos archivos EPUB y un MOBI validados**.

Archivos de entrada

Todos los archivos están disponibles aquí.

Como hemos procedido en los otros métodos, nuestros archivos iniciales —ubicados en `don_quijote/desde_cero/archivos_madre`— son los siguientes:

* `todo.xhtml`. El contenido del libro.
* `portada.jpg`. La portada del *Don Quijote*.
* `principal.css`. Los estilos que usaremos para esta edición.

Estos son los archivos madre, los cuales ya no tenemos necesidad de modificar porque ya le hemos dado el formato adecuado. Nos ahorramos ese dolor de cabeza, donde el cuidado en **el formato es esencial para el control editorial de cualquier publicación, impresa o digital**.

¡Manos a la obra!

Ubicados en `don_quijote/desde_cero` solo usamos el comando `pc-automata` para llamar a `automata.rb` que, a su vez, llamará al resto de los *scripts* y los programas:

```
nika-zhenya@archlinux:~/Descargas/don_quijote/desde_cero
nika-zhenya en ~/Descargas/don_quijote/desde_cero
$
```

Producción de dos EPUB y un MOBI en minuto y medio.

1. Creación del proyecto EPUB

El primer comando es para crear una carpeta que será donde las herramientas trabajen:

```
pc-automata --init
```

Esto nos crea una carpeta llamada `epub-automata`, el cual solo contiene dos ficheros:

- `.automata_init`. Un archivo oculto que permite validar la carpeta como un directorio para lo que hará `pc-automata`.
- `automata_meta-data.yaml`. Un archivo donde pondremos los metadatos del libro.

2. Agregación de los metadatos

El siguiente paso es abrir el archivo `automata_meta-data.yaml` para introducir los metadatos. Este fichero puede abrirse con cualquier programa; por ejemplo, Bloc de Notas, TextEdit o Gedit. Utilicé nano, un editor de texto que se usa desde la terminal:

```
nano epub-automata/automata_meta-data.yaml
```

3. Creación de los EPUB y el MOBI

Con nuestro proyecto y los metadatos listos, solo es necesario escribir este comando:

```
pc-automata -f archivos_madre/todo.xhtml -c archivos
```

Pero ¿qué significa todo eso? Vayamos por partes para observar que no es nada complejo:

- `pc-automata`. Con esto se manifiesta que llamaremos a `automata.rb` para producir los EPUB y el MOBI de manera automatizada.
- `-f archivos_madre/todo.xhtml`. Este primer parámetro señala cuál es el archivo (*file*) que contiene todo el libro.
- `-c archivos_madre/portada.jpg`. Con este otro indicador se menciona cuál es la portada (*cover*) de la obra.
- `-s archivos_madre/principal.css`. Aquí se enlaza la hoja de estilos (*style sheet*) de la publicación.
- `-d epub-automata/`. Si no estamos dentro de la carpeta (*directory*) del proyecto, se tiene que indicar dónde se encuentra.
- `-y epub-automata/automata_meta-data.yaml`. Como no estamos dentro del directorio, también hay que señalar dónde está el archivo YAML que contiene los metadatos.
- `--section`. Por último, se indica que la división del documento no se hará cada etiqueta `<h1>` sino cada `<section>`.

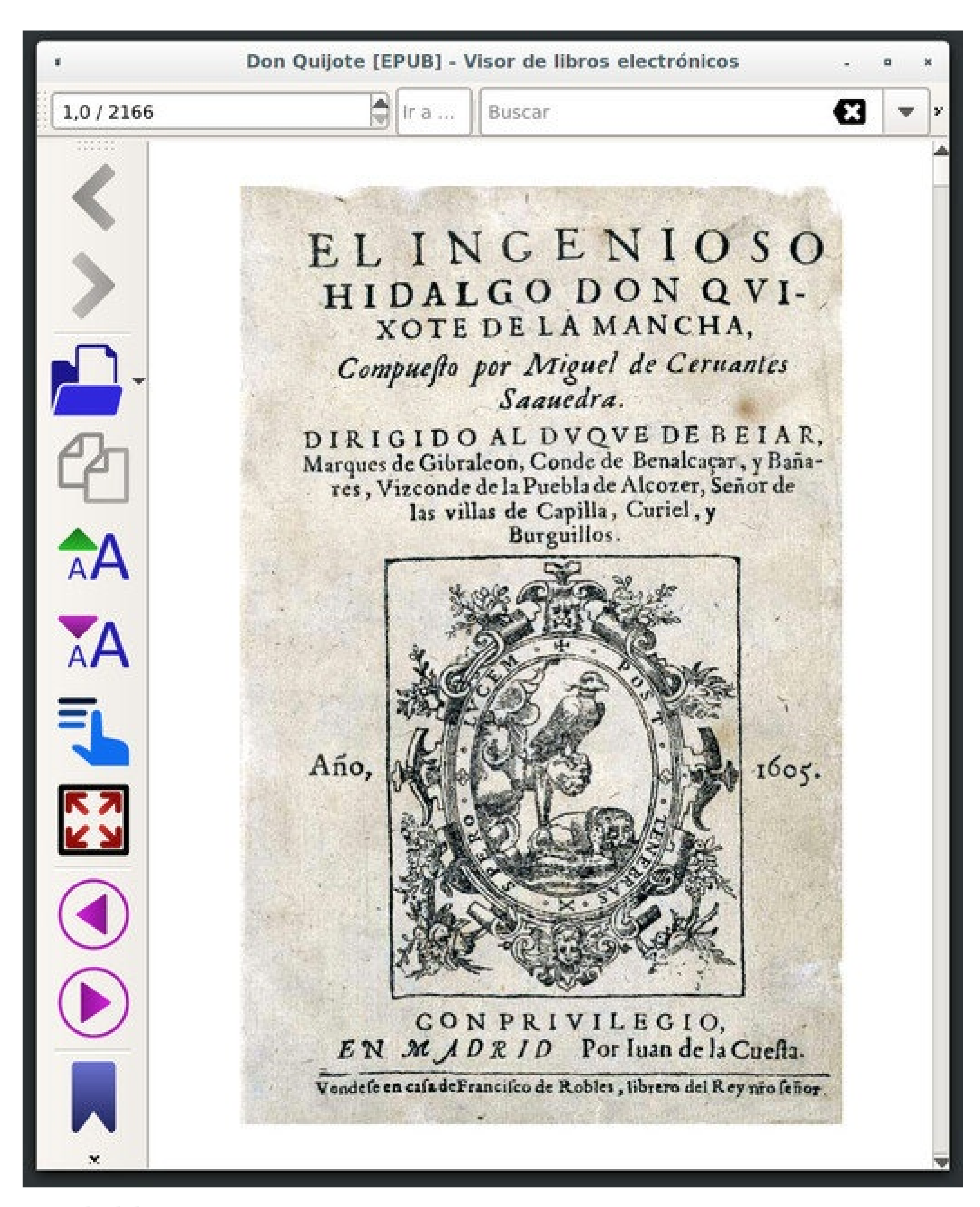
Con esto —como podemos ver en `don_quijote/desde_cero/epub_automata/log.txt`— `pc-automata` lleva a cabo estas tareas:

1. Crea un proyecto EPUB con `pc-creator`.
2. Divide el archivo `todo.xhtml` en cada etiqueta `<section>` con `pc-divider`.

3. Crea el EPUB versión 3.0.1 —todavía no lo actualizo a 3.1— con `pc-recreator`.
4. Crea el EPUB versión 3.0.0 a partir del EPUB 3.0.1 con `pc-changer`. La versión 3.0.0 es la más popular y la aceptada por terceros, como Amazon, iTunes y Google Play.
5. Se verifica el EPUB 3.0.1 con `epubcheck`, la herramienta oficial de validación que es mantenida por el International Digital Publishing Forum.
6. Se verifica el EPUB 3.0.0 también con `epubcheck`. Si hay errores, aparecerán indicados en la terminal y quedará guardados en el archivo `log.txt`.
7. Si se cuenta con `kindlegen`, termina los procesos creando el MOBI a partir del EPUB 3.0.1.

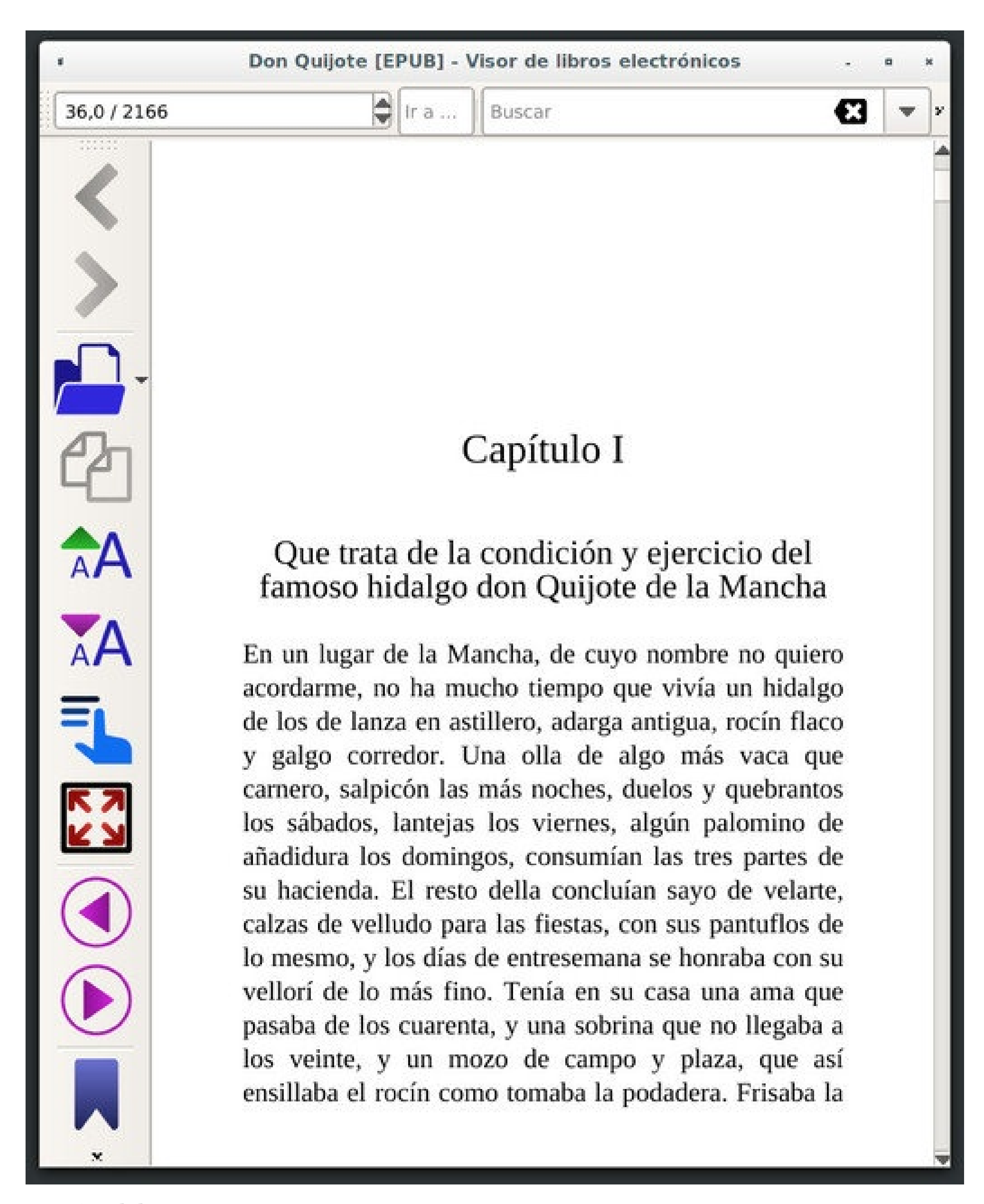
Primer resultado

Con esto se crean los EPUB y el MOBI cuyo resultado es:



Portada del EPUB.

Como puede verse, los estilos están bien aplicados:



Interior del EPUB.

Y, por supuesto, automáticamente se creó la tabla de contenidos:

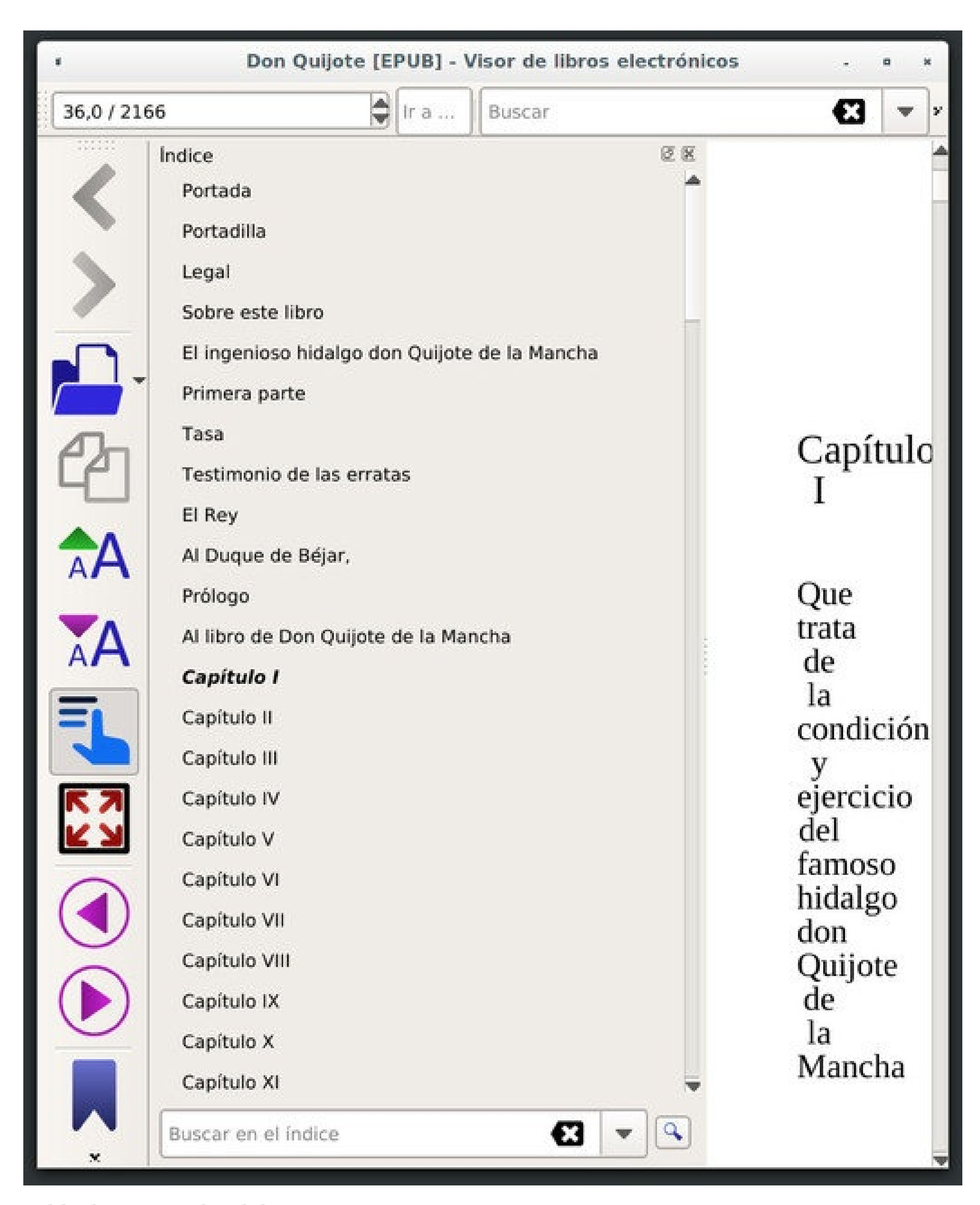


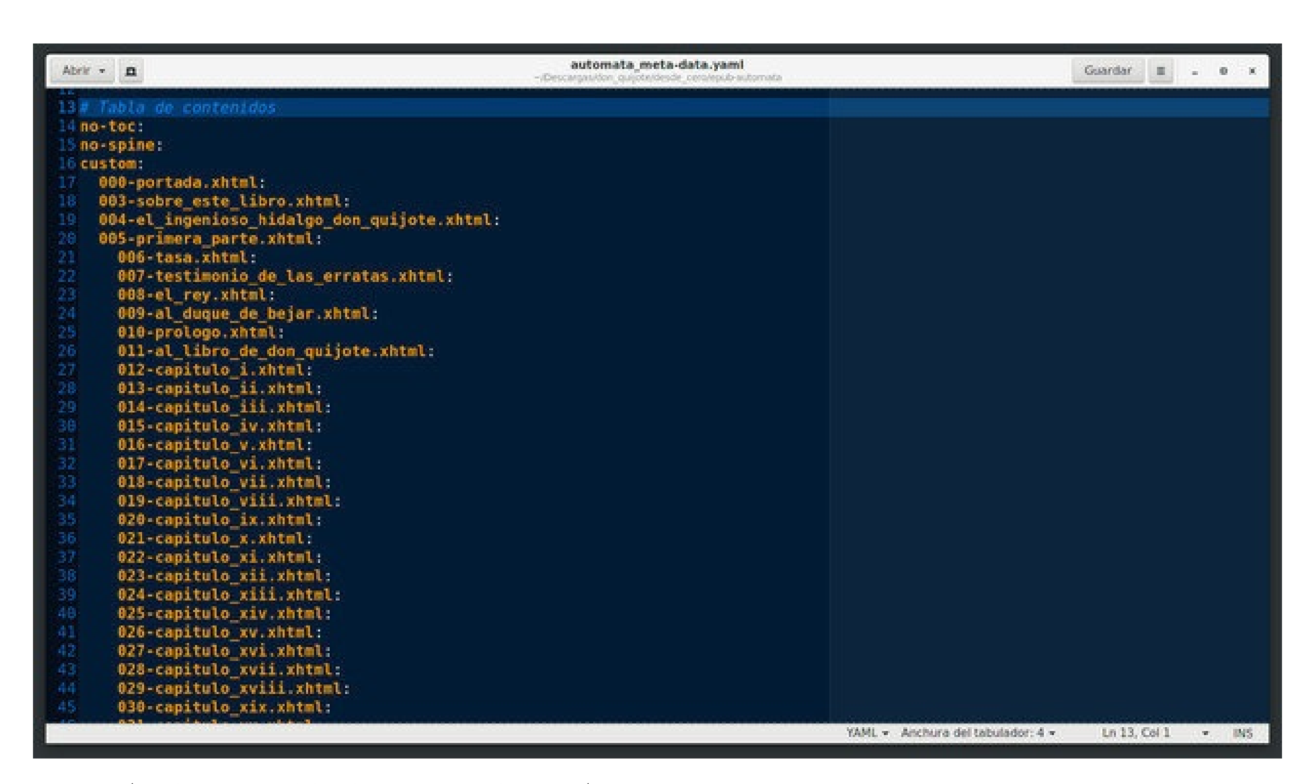
Tabla de contenidos del EPUB.

¡Momento! En la tabla de contenidos todas las secciones tienen la

misma jerarquía… Por experiencia, esto no incomoda a la mayoría de las personas. Sin embargo, hay ocasiones donde se quiere ocultar elementos —como la página legal— o crear una tabla de contenidos jerarquizada.

No hay ningún problema, es solo cuestión de modificar nuestros metadatos. Si se abre de nuevo el archivo `automata_meta-data.yaml` se observa el área para la tabla de contenidos, donde la opción más personalizable es el área de `custom:`.

Desde ahí acomodamos los elementos que deseamos mostrar en la tabla, añadiendo espacio a inicio de cada línea, unos más que otros, para jerarquizarlos:



Creación de una tabla de contenidos jerárquico.

Segunda recreación

Con la tabla de contenidos modificada, solo es necesario volver a

correr el comando para producir los archivos. La sintaxis es la misma a cuando desarrollamos los formatos por primera vez:

```
pc-automata -f archivos_madre/todo.xhtml -c archivos
```

El proceso vuelve a comenzar sin ninguna dificultad:

nika-zhenya@archlinux:~/Descargas/don_quijote/desde_cero
nika-zhenya en ~/Descargas/don_quijote/desde_cero
$

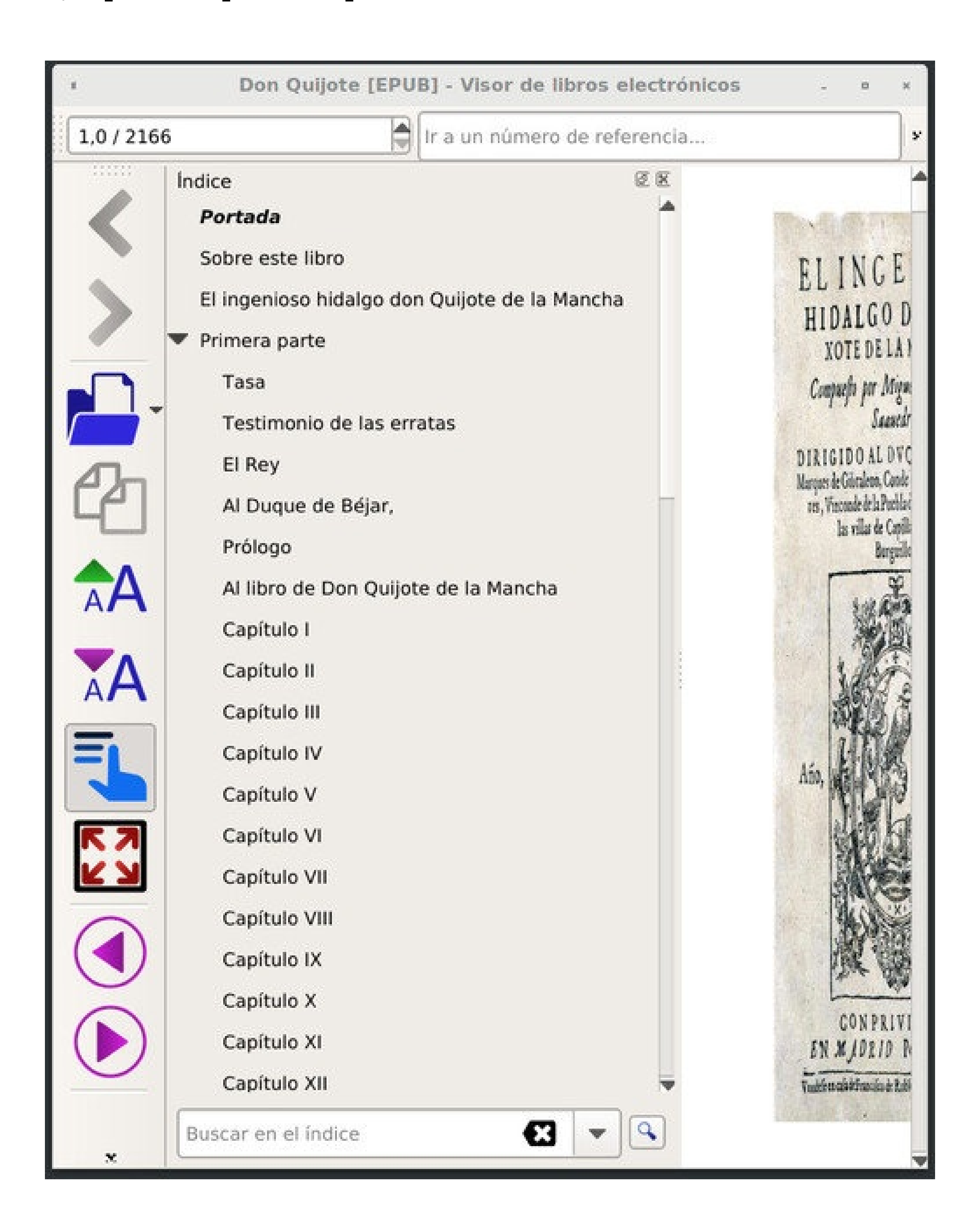
Nueva producción de los EPUB y el MOBI.

¡Ya tenemos de nuevo nuestros libros! Con una hermosa tabla de contenidos:



EPUB con una tabla de contenidos jerárquica.

Que puede expandirse para ver el resto de las secciones:



EPUB con una tabla de contenidos expandida.

Dos curiosidades

Como se observó en la segunda recreación, el proceso de automatización sigue una idea de desecho: todos los archivos que no sean archivos YAML son eliminados para volver a generar todos los ficheros.

¿Por qué? Por un lado, porque la generación de archivos puede llegar a confundir al usuario si se mezclan con ficheros antiguos. Por el otro, la idea metodológica es el trabajo con los archivos madre, así que, en caso de modificación, ¿por qué se querría utilizar los ficheros generados en lugar de modificar los archivos madre?

Con esto **es posible no preocuparnos por los formatos finales**, a tal grado que incluso no es necesario subirlos al servidor, ya que en cualquier momento se pueden volver a generar a partir de los archivos madre.

Así nos ahorramos un poco más de espacio y nos quitamos la preocupación de estar respaldando los archivos finales a cada instante. El control es tal que la mala práctica de tener los últimos cambios en los archivos finales no tiene cabida.

¿Es necesario hacer modificaciones que con la automatización no se pueden llevar a cabo? Aunque se trata de una singularidad, la posibilidad no queda eliminada, ya que `pc-automata` conserva el proyecto EPUB sin compresión en la carpeta `epub-automata/epub-creator`.

En la mayoría de los casos no se recomienda trabajar con esos archivos, pero la flexibilidad permanece porque entre menos control hay en un proyecto editorial, más lugar hay para las

excepciones…

¿Quieres probar Pecas en modo automatizado?

Si tienes curiosidad de probar Pecas como se hizo aquí, puedes recrear este ejercicio. Solo es necesario:

- Descargar los archivos del Don Quijote (elimina la carpeta `epub-automata`).
- Instalar todo lo necesario para Pecas.

También existe otro ejercicio adicional en el cual puedes crear los EPUB y el MOBI desde un archivo PDF y pasando por Markdown. Para acceder, solo haz clic aquí.

Sobre esta entrada

- Título original: «De XHTML a EPUB y MOBI ¡en solo minuto y medio!».
- Título en el *blog* de Mariana: «De XHTML a EPUB y MOBI con Pecas, ¡en solo un minuto y medio!».
- Fecha de publicación: 8 de diciembre del 2017.
- Redacción original.
- Revisión de Mariana.

III. Gamificación de la edición

Appbooks y realidad aumentada, ¡no todo son ebooks!

Las posibilidades tecnológicas actuales permiten una clase de publicaciones cuya metodología poco tiene que ver con la tradición editorial. Esta esencial diferencia se debe a que se trata de libros electrónicos que sobresalen por su alta interactividad o propuestas que no se ciñen al paradigma de la página, la lectura lineal de una obra y las narrativas que permite este soporte. Este tipo de **productos editoriales** es lo que últimamente se conoce como ***appbooks***.

Este término proviene principalmente por la popularización de las tiendas de aplicaciones como Google Play o App Store. Sin embargo, **los *appbooks* anteceden a los teléfonos inteligentes y las tabletas**.

De hecho, tienen su antecedente cuando se empezó a experimentar con la escritura a través de la computadora y cuyos resultados no se pensaron para la impresión.

Por este motivo, los *appbooks* han existido no solo como aplicaciones para iOS o Android, sino también en forma de CD interactivos (¿quién no recuerda la enciclopedia interactiva de Salvat?), plataformas *web*, videojuegos o aplicaciones con realidad aumentada.

Metodologías de desarrollo variable según el tipo de narrativa buscada

Para los *appbooks* no existe un programa especialmente diseñado

para su creación. En su lugar, se utilizan una serie de diferentes tecnologías:

- **Páginas HTML estáticas** en los que únicamente se usa HTML, CSS y JavaScript. Estas son una de las propuestas más tempranas que se basaron, principalmente, en la narrativa hipertextual (donde la lectura no tiene un seguimiento lineal, sino que da «saltos» a través de enlaces *web*). Aunque no se reduce únicamente a esto, sino también al empleo de programación para cambiar de manera dinámica el texto o desatar otros eventos, como en CLIteratura.
- **Aplicaciones *web*** que interactúan con un servidor para obtener información. La originalidad narrativa de estas aplicaciones tiende a ser su manera de organizar una gran cantidad de información textual. Por lo que en general se tratan de diccionarios, como *Stanford Encyclopedia of Philosophy*; enciclopedias, como Wikipedia; o catálogos, como el que en Nieve de Chamoy estamos creando para las obras del Museo Soumaya.
- **Aplicaciones híbridas** en las que se mezclan elementos *web* con el uso del *hardware* del dispositivo móvil. Esto hace posible que el *appbook* no tenga que abrirse desde un explorador, sino que esté alojado en el móvil como una aplicación independiente. En este, la narratividad aún está centrada en lo textual, al menos al momento de organizar la información, por lo que el texto se estructura de tal manera que es sencillo leer y navegar a través de tantos datos. Ejemplos son la *app* de Jazz at Lincoln Center o la aplicación que hicimos sobre los cien años del Futbol Club Atlas.
- **Aplicaciones nativas** que utilizan directamente las

posibilidades que ofrece el dispositivo. Estas aplicaciones se desarrollan así para obtener una gran optimización para una narrativa interactiva, como *Our Choice* o la *Guía del Prado*.

- ***Software* creado con motores de videojuegos**. Las posibilidades narrativas se aproximan más a la *jugabilidad* y, por lo general, ya contemplan ciertas características habituales de los videojuegos, como un sistema de puntuación o la interacción con personajes virtuales. Esto es perceptible en el trabajo que realiza Concretoons, los videojuegos independientes como *The Mammoth* o la aplicación *En busca de Kayla* que desarrollamos para la editorial Sexto Piso.
- ***Software* que involucra realidad aumentada.** Esta clase de *appbook* busca alimentar la narrativa textual de una publicación impresa. Ofrece elementos multimedia que se desatan cuando se enfoca la cámara del dispositivo sobre la obra, como *La leyenda del ladrón* o la que desarrollamos para la Universidad Nacional Autónoma de México.

Limitaciones temporales, técnicas y económicas

En cualquier ámbito profesional o institucional de trabajo siempre tenemos el problema de que, por lo general, el tiempo destinado a un proyecto no es el deseado para el resultado más óptimo. Algunos detalles no son perceptibles para el público en general; por ejemplo, los callejones blancos de un impreso, que sí tienden a ser vistos entre editores. Es decir, muchas veces esos *errores* no afectan al uso general de un producto.

Sin embargo, **en el desarrollo de *appbooks* los descuidos son lo**

primero que advierte el usuario. ¿Cuántas veces ha molestado usar una aplicación cuya navegación no es «amistosa» o hace cosas «inesperadas»?

Los detalles que pueden encontrarse en un *appbook* suelen afectar a la navegación general de la obra, lo que puede llevar a una evaluación negativa o devolución del producto.

En la creación de *appbooks* ha de tenerse presente que estos detalles estarán presentes si el tiempo es escaso. Esto es relevante si la publicación es para un tercero, ya que se ha de indicar para evitar futuros problemas.

En este sentido, es ideal que los *appbooks* cuenten con tiempos de pruebas y de optimización según los dispositivos de salida. Estos tiempos pueden ser de algunas semanas a varios meses, según la complejidad de la publicación.

«Fase de pruebas» y «fase de optimización» no son procesos comunes dentro de la producción del libro: **los *appbooks* tienen una metodología más afín al desarrollo de *software* o de videojuegos**. Por lo general, esto también implica limitaciones:

- quienes editan carecen de conocimiento para el desarrollo de *software*, por lo que el *appbook* presenta problemas de desempeño, o
- quienes programan desconocen el control de la calidad editorial, ocasionando que el *appbook* tenga una baja calidad en la edición.

La **recomendación** para los que desean crear este tipo de publicaciones es que, en un primer momento, se decidan por un tipo de *appbook* de los que se mencionaron en la sección anterior y que, de manera paulatina, se vaya aumentado el grado de

complejidad. Pero, y principalmente, que se tenga en cuenta que la inclusión de un artista digital y alguien que programe es indispensable.

Y considerar que cuando se realiza una contratación externa de estos servicios existe la posibilidad de una falta de comunicación, ya que la jerga editorial les es desconocida, así como los conceptos fundamentales para el desarrollo de *software* o del arte digital no forman parte de la cultura general de quien edita, lo que puede ir en detrimento de la calidad.

Estas limitaciones de tiempo y de técnica acarrean que los costos de producción de un *appbook* no sea el que se piensa en un principio, más si no existe una planificación adecuada. ¡Se está desarrollando *software*!

El costo de creación de *appbooks* es igual o mayor al de un impreso. Si se piensa que el desarrollo de un *appbook* implica tiempos y esfuerzos similares al desarrollo de un libro electrónico estándar (EPUB, MOBI o IBOOKS) es mejor que este entusiasmo se guarde para un proyecto muy importante…

El lugar actual de los *appbooks* en el mundo editorial

Por la disparidad metodológica y las diversas limitantes que implica el desarrollo de aplicaciones, **los *appbooks* no han llegado para reemplazar a los libros**, ni a los *ebooks*, al menos no durante los siguientes cinco o diez años. El lugar de esta clase de publicaciones no es dentro del grueso de la producción de libros, sino de excepciones dentro del mundo editorial.

Cuidado, que este carácter marginal no nos engañe. Esto no quiere decir que su potencial comercial sea escaso. Al contrario, un *appbook* bien desarrollado y publicitado puede fácilmente

transformarse en un contenido viral que genere millones de descargas.

Sin embargo, una alta calidad en el desarrollo y una buena estrategia de *marketing* digital, orientada a la venta de aplicaciones, no son campos dominados por el sector editorial. El sector del desarrollo de *software* tampoco tiene un completo dominio, ya que en su mayoría se orientan a la creación de sistemas y soluciones empresariales, no a la creación de aplicaciones interactivas para el público general.

Los estudios de videojuegos son los que **tienen una mayor ventaja para el desarrollo de *appbooks***, debido a que son los que cuentan con un alto dominio y experiencia de las características técnicas y de las narrativas interactivas necesarias para interesar al usuario. De hecho, esto queda fácilmente demostrado al ver quién está detrás de nuestros *appbooks* favoritos.

Entonces, ¿qué hacemos en la industria editorial con los *appbooks*? ¡Experimentar! **Los *appbooks* pueden ser una muestra sobre cómo será posible «deconstruir» la lectura** y, por ende, una evidencia de los profundos cambios que representan las publicaciones digitales para los procesos editoriales.

Esto se debe a que recuperar la inversión necesaria para el desarrollo de un *appbook* es aún más difícil que agotar una edición de un impreso. Por ello, es mejor tomar el proyecto con cierto pesimismo y partir del supuesto de que la inversión no se recuperará.

O bien, crear esta clase de publicaciones si la entidad editorial cuenta con la posibilidad «orgánica» de producción, donde todo se crea internamente con una clara idea técnica y un calendario realista.

Otra opción, que también implica tener una estructura

orgánica, es ofrecer los *appbooks* como un servicio editorial a instituciones públicas o privadas, para ir ganando experiencia. Una destreza que servirá al momento de producir *appbooks* bajo nuestro sello editorial.

Por otro lado, en el mundo editorial todavía es muy difícil separar la idea de «obra» a la de «libro», más cuando este último término quiere decir «libro impreso». La asociación había sido tan normal que el simple hecho de hablar de diseño fluido, como es posible en libros electrónicos estándar (EPUB, IBOOKS, MOBI), ya es una noción difícil de entender porque cambia la idea de lo que se entiende por «página». A esta complicación sumemos que en muchos *appbooks* la idea de «página» no es difusa sino inexistente…

Paradigma de edición para *appbooks*: ¿páginas o dinámicas?

Ejemplos de esta dificultad los tenemos en Nieve de Chamoy todos los días. Cada vez que hacemos una aplicación para un cliente —en muchos casos profesionales de la edición—, lo primero a esclarecer es que **en el desarrollo de los *appbooks* es indiferente la cantidad de páginas**. Para su creación lo relevante es la cantidad de dinámicas que se podrán hacer, los recursos que se tienen y la cantidad de imágenes, videos o audios a incluir (e incluso también el formato de origen de los contenidos).

La complejidad no es proporcional a la cantidad de texto, como lo es en la corrección o la traducción. La «página», la «cuartilla» o las cantidades de palabras o de caracteres son tan fundamentales para calcular los costos en el mundo editorial que, por lo general, la creación de un presupuesto para *appbooks* ya es una causa de conflicto.

Además, si entre quienes editamos la publicación de libros electrónicos estándar es un reto, mayores dificultades enfrentamos en el desarrollo de *appbooks*. Si entre los profesionales de la edición aún es muy difícil desembarazar las ideas de «obra» o «libro» de la de «impreso», o de que **la página ya no es el único paradigma para la edición**, ya podemos imaginarnos lo complicado que esto puede ser para el lector en general que, sin dudas, estimará o rechazará la propuesta narrativa de un *appbook* dentro de un universo digital donde se convive con las películas y los videojuegos.

Para finalizar, comparto una pequeña lista de reproducción de *appbooks* de diversa índole. Este material lo empleamos en talleres, seminarios o pláticas. Si conocen un video que sea relevante para esta lista, por favor, deja constancia de él en los comentarios.

Sobre esta entrada

- Título original: «*Appbooks* y realidad aumenta, ¡no todo son *ebooks*!».
- Título en el *blog* de Mariana: «Appbooks y realidad aumentada. ¡No todo son ebooks!».
- Fecha de publicación: 22 de marzo del 2017.
- Redacción original.
- Revisión de Mariana.

El videojuego y el libro: la gamificación de la edición

La tecnología digital ha traído dos fenómenos al mundo de la edición:

1. la automatización, el control y el mejoramiento editorial mediante el uso de *software*, y
2. la apertura de la noción de «libro» más allá de lo que los lectores y los editores conocieron durante cinco siglos.

El primer fenómeno tiene su máxima expresión en la publicación estandarizada, principalmente en el ámbito académico. El segundo caso nos lleva a **los límites de la tradición editorial: el videojuego.**

El extraño caso de una alianza y un antagonismo

Entre los muchos de los interminables debates dentro del mundo de la edición —como si la publicación digital viene a *sustituir* al impreso— hay un tema que de manera constante también es motivo de discusión: «el libro o el videojuego, ¿de qué lado estás?».

El libro, no como este u aquel objeto, sino como idea y como tradición que atraviesa y tiene raíces profundas en nuestras cultura, es tanto un medio como un fin. El libro se ha **entendido como un vehículo de entretenimiento, pero también de transmisión de conocimiento** e incluso de sabiduría.

Al mismo tiempo, el libro ha representado una de las formas más acabadas de expresión cultural, o al menos como el cierre de

los complejos actos de la creación literaria o la expresión de una idea.

El libro es tanto ese objeto que adquirimos, prestamos, maltratamos, sacamos fotocopias, hasta que del libro no queda nada sino puras hojas sueltas, como también ese ideal donde se constata el cumplimiento de una actividad literaria, filosófica o científica. Cuando decimos «libro» no solo pensamos en letras y papel, sino también en la idea de cumplimiento.

Entre las múltiples posibilidades de la tecnología digital, unas de ellas han empezado a visibilizar e incluso diluir la idea de la completud de una obra o la definición de lo que es un «libro».

En otra ocasión nos ocuparemos de la idea —inconveniente, según mi opinión— de la completud. **El libro ahora no puede entenderse como texto y papel**, porque ni todos los libros han tenido texto ni todos están compuestos de papel, **y, principalmente, debido a que hay libros cuya comprensión es mediante el juego**.

Hay libros cuya experiencia no es tanto mediante la lectura o la observación de imágenes, sino a través de una posición activa por parte del lector para solucionar alguna clase de desafío o, simplemente, para darle continuidad a la trama de la obra. Estos elementos están presentes en el videojuego.

Con esto no se quiere decir que el videojuego trajo consigo estas posibilidades, ya que el juego y, especialmente, el carácter lúdico de un libro es anterior a ello —piénsese en los libros de sopas de letras o en *Rayuela*, de Cortázar—. Sin embargo, es con el videojuego donde encontramos una de sus expresiones más radicales.

Por desgracia, esta semejanza y posibilidad que trae el videojuego para generar otras experiencias de lectura, por lo general, se ve opacada por el contacto que la mayoría de los lectores

han tenido con los videojuegos.

Si se compara la narrativa de alguna franquicia como *Mario Bros, Call of Duty, Grand Theft Auto* o *Minecraft*, un lector poco o nada podrá encontrar interesante. ¿Qué valor tiene el rescate de una princesa por parte de un fontanero si existen Don Quijote y Dulcinea?

Un juicio valorativo hacia el videojuego basado en las franquicias que más ganancias han generado es como emitir una opinión sobre la cultura letrada basándose en los libros más vendidos, que al menos en el caso mexicano son los libros de autoayuda y los libros de texto.

En este sentido, quizá lo más pertinente es pensar el videojuego desde otros ejemplos concretos, como *Journey*, *Braid*, *The Unfinished Swan*, *Limbo*, *Machinarium* o *A Noble Circle*.

Del lápiz al papel; el teclado y los gestos, la realidad virtual y más allá

Llenar un crucigrama o jugar *Calabozos y Dragones* (*Dragones y mazmorras*, en España) es una de las primeras experiencias lúdicas que se tienen gracias a alguna clase de libro, aunque solo sea un instructivo.

A la par de la idea de «libro» como pilar de nuestra cultura, también está patente su concepción como un **producto más de entretenimiento**, a tal punto que desde sus comienzos se ha experimentado para que este carácter lúdico permita una mayor actividad del lector.

Mundos ficticios y Don Quijote

Con el advenimiento de las computadoras personales y el internet se posibilita **tener mayor participación ya no como lector, sino también como jugador**.

Uno de los ejemplos que más me agradan son los MUD, mundos ficticios (por lo general, ficción medieval) construidos con puro texto y cuya interacción es escribiendo instrucciones con un teclado.

Del entretenimiento que puede encontrarse en *Don Quijote* al hacer sátira de las novelas de caballerías, en el MUD el jugador puede ser el lector de su propia historia inserta en un mundo tan artificial como es uno cuya representación es mera tipografía digital.

Si en uno nos entretiene las aventuras de Don Quijote, en el otro el usuario disfruta de ser él mismo Don Quijote.

Machinarium, Braid, Limbo y la magdalena de Proust

El *mouse* sintetiza una serie de comandos escritos en un teclado en simples clics, *scrolls* y *drags*. La experiencia de **un libro** también **permite otro nivel de interactividad**: las **aventuras *point-and-click*** donde se interacciona con objetos en una escena para darle continuidad a una historia.

Uno de mis ejemplos favoritos es *Machinarium*, donde un robot se adentra en una ciudad de robots para terminar descubriendo lo que es la amistad en un mundo tan miserable, y que hasta cierto punto recuerda a *El principito* y su continua búsqueda.

Una mezcla del uso del teclado y el ratón también permite otro tipo de interacción con el lector-jugador, especialmente en contextos gráficos y en «tiempo real». Solucionar una serie de problemas es una de las formas más básicas de interactividad, como puede verse en *Braid* o *Limbo*.

Aunque parezca sencillo, su solución requiere de ingenio hasta que al final del juego se percibe una tendencia existencial: el pasado que se ha sido, como en *Braid*, o el desconocimiento de no saber si se está vivo o muerto, como en *Limbo*.

Entre la nostalgia al pasado provocada por una magdalena en *En busca del tiempo perdido* con Proust y la mecánica de regresión del tiempo en *Braid* existe un gran abismo. Sin embargo, la sensación al leerlo y al jugarlo es muy similar: el retorno es posible aunque no puede modificar el tiempo presente.

Mundos abiertos: Journey, The Unfinished Swan y Oliver Twist

La exploración de mundos abiertos; es decir, de entornos donde no es necesario cumplir la historia como una receta, sino según como se antoje, es otra de las experiencias posibles con un teclado y un ratón, y claro está, con un mando.

Journey o *The Unfinished Swan* son buenos ejemplos. En el primero se habita en un mundo posapocalíptico que no causa terror sino desasosiego. En el segundo un niño huérfano empieza a seguir un cisne, lo cual lo llevará a conocer sobre su historia y la de sus padres.

Existe cierto paralelismo entre *Oliver Twist* y *The Unfinished Swan*, ya que ambos son niños huérfanos que por algún u otro motivo se ven insertos en una ciudad que es un acertijo. Pero también está la gran diferencia en que en la ciudad de *The Unfinished Swan* no hay ni un solo habitante, cuyo medio para aprender a moverse entre vericuetos es aventar globos rellenos de agua o de pintura.

A partir de este punto, **la experiencia lúdica o lectora empieza a ser más experimental y fragmentaria**. No porque no sea posible

«contar una historia», como tradicionalmente lo entendemos, sino porque el *hardware* disponible es relativamente reciente y faltan algunos años para que una persona amante de los libros también tenga el control técnico sobre estas herramientas.

Pantallas táctiles y realidad virtual o aumentada

El caso más inmediato son los gestos con los dedos. Las pantallas táctiles permiten otro tipo de interacción donde el intermediario entre la computadora y nosotros no es una herramienta que pueda tomarse con las manos, sino un dispositivo que puede interpretar nuestro movimiento con los dedos.

Un ejemplo narrativo que también genera una experiencia lectora es *A Noble Circle*, donde el jugador es un círculo que mediante toques con el dedo va saltando para evitar obstáculos pero, al mismo tiempo, un narrador te va explicando el porqué este círculo tiene la «necesidad» de saltar, concluyendo en la descripción de un mundo compuesto por figuras geométricas y sonidos.

Las tecnologías de reconocimiento, además de potenciales problemas de privacidad, también permiten otra interacción que en determinado punto puede permitir otras experiencias de lectura.

Desde el reconocimiento facial, hasta el reconocimiento corporal y del entorno físico, como está presente en la **realidad virtual o aumentada**, hay un enorme campo para la experimentación que el mundo editorial recién está empezado a tomar en cuenta, con especial atención a la realidad aumentada, debido a la posibilidad de mostrar contenidos multimedia a través de un impreso.

¿Es posible leer sin ver?

Pero no nos quedemos ahí, existe **otro tipo de interfaz aún no comercial** que puede tener un fuerte impacto no solo en la industria del entretenimiento, sino también en la del libro: **la interfaz cerebro-máquina**.

El usuario se coloca un gorro que tiene una red de nodos que miden la actividad cerebral. A partir de esta actividad, el usuario tiene la posibilidad de interactuar con computadoras o con máquinas.

Ya se han hecho pruebas jugando *World of Warcraft*. Por tanto, ¿qué pasaría si un día la experiencia lectora no necesita nada más que nuestra mente conectada a una máquina?

Para atenuar un poco, no se habla tanto de la Matrix, sino de la forma en cómo hemos accedido al contenido de una obra.

Las pantallas y la flexibilidad de la publicación digital están revolucionando el mundo editorial, al punto de destruir la estaticidad de la «página» o la definición de un «libro» acorde a sus propiedades físicas. Entonces, ¿qué podemos pensar cuando ya ni siquiera la vista sea necesaria para «leer», cuando la «lectura» ya no sea definible por una actividad ocular?

Quizá el camino que está abriendo el **audiolibro**, al desafiar la idea de la vista como medio predilecto para el acceso a un libro, sea solo la punta del *iceberg* de lo que se avecina en el futuro…

Más presente y menos futuro

En lo personal no me gusta indagar sobre el «futuro» del libro, principalmente porque lo que en muchos foros se trata como un

tema futurista es más bien algo que ya está ocurriendo, pero que se desconoce o, peor aún, no se percibe su potencial importancia.

La gamificación de la edición es un ejemplo de ello. Durante ya mucho tiempo se ha estado gestando la interrelación entre el libro y el videojuego. Sin embargo, la mayoría de las veces estos experimentos fueron etiquetados como «videojuegos» o, en el mejor de los casos, como un «estudio de caso».

O bien, sus ejecuciones, al desarrolladas por estudios independientes de videojuegos y no por grandes compañías con mucho presupuesto para publicidad y desarrollo, quedan fuera del panorama de un amante de los libros. Los ejemplos empleados aquí son solo una pequeña muestra de cómo el desarrollo independiente de videojuegos es una de las principales ramas que se acerca al discurso o tramas literarias.

Una plataforma para conocer más de este tipo de videojuegos es Itch, por si gustan visitarla, sus videojuegos son gratuitos o de bajo precio.

Pienso que si han sido etiquetados de esta forma es porque muchos de estos experimentos no cumplen con lo que de manera tradicional se ha entendido por «libro» ni a muchos de sus ejecutantes les interesa que sus obras sean catalogadas de esta manera.

Pero también es debido a que **el editor promedio no le interesa nada que vaya más allá del horizonte de su profesión**. Son pocas las personas en el mundo de la edición que están al tanto de lo que se puede aprender al jugar videojuegos.

Entre estas, son menos las que *efectivamente* el videojuego no solo se constituye como una forma de entretenimiento sino como una cultura. Todavía más reducido es el grupo de personas que, a su vez, tiene los medios técnicos para dejar de ser un espectador y

meterse de lleno con la experimentación de estas posibilidades.

Así que, antes de imaginarnos experiencia de lecturas tipo Matrix, ¿por qué no mejor probamos con no solo leer, sino también jugar un videojuego?

¿Por qué no empezamos a ver al videojuego no como un objeto de entretenimiento o como un producto infantil, sino **como una creación cultural que** en su dinámica nos **permite repensar y recrear lo que durante cinco siglos conocimos como «libro»?**

Sobre esta entrada

- Título original: «Videojuegos y libros, la gamificación de la edición».
- Título en el *blog* de Mariana: «El videojuego y el libro: la gamificación de la edición».
- Fecha de publicación: 17 de octubre del 2017.
- Redacción original.
- Revisión de Mariana.

CLIteratura, un proyecto sobre código (de programación) y literatura

¡No todo es publicación estándar! Lo más llamativo de las tecnologías digitales en relación con el mundo del libro es, sin duda, la gamificación de la edición. Desde los *appbooks* hasta la realidad aumentada o virtual, en diversos eventos relativos al libro se anuncian estos productos como el «futuro» del libro.

Aunque más bien es el «presente» de las posibilidades de la edición digital, muchos de estos productos tienen una relación muy definida con las capacidades técnicas digitales que vale la pena resaltar.

El código como infraestructura

Lo más llamativo de este tipo de publicaciones no estandarizadas —no utilizan los mismos procesos, las técnicas varían y hay casi tantas metodologías como productos posibles— es el dinamismo y la «pérdida» de estructura. **Lo que se pierde no son conocimientos tradicionales de cuidado editorial, sino una noción que era ley en el reino exclusivo del impreso: la página**.

La página limitaba las posibilidades a la publicación, pero también era un lienzo que daba fuentes ilimitadas de creatividad literaria, tipográfica y gráfica. La publicación estandarizada hace una metáfora de esta noción, pero en los procesos no estandarizados de edición digital la página en muchas ocasiones es sustituida por la **idea de escena**.

En una escenificación hay dinamismo: el diálogo y la escenografía se orientan a una narrativa que no espera a ser leída. La historia avanza y el lector se hace espectador y, a veces, actor.

En el caso de las escenas planificadas para la producción de *appbooks*, ya se busca que el espectador sea un agente constante dentro de la misma narrativa. Sea desde una simple navegación interactiva hasta el completo sumergimiento en la historia, el espectador se hace usuario.

Y en este **traslado del lector al usuario del libro** lo que resalta son el tipo de recursos a los que se tiene acceso y cómo estos son presentados y modificados por el usuario.

Sonidos, videos, animaciones, física simulada, análisis de datos, uso de gestos o creación de mundos tridimensionales: la edición engloba ya más elementos. *Editores ¿pueden con esta tarea?*

El código: pilar para la edición digital

Lo anterior no sería posible sin una base informática que permita el uso de dichos recursos. **El código viene a ser el pilar para la edición digital** y no hay caso más evidente de esta necesidad que las publicaciones digitales no estandarizadas.

Sin el código no hay manera de publicar estos y casi todos los productos editoriales. O si somos un poco severos: **la edición ha quedado subordinada al código**. ¿No lo crees así? ¿Usas una computadora para editar o publicar tus libros? Bienvenido, el código te respalda.

Más allá de las limitaciones del programa o de su uso, rara vez se vuelve un tema de conversación la dependencia tecnológica de la edición. De manera inexplicable o mera renuencia, el código se percibe como dado.

A nosotros, editores, ya se nos ha otorgado el código en su forma más amigable: como un programa con interfaz gráfica, el código compilado en archivos ejecutables. Por el código la edición se ha transformado, para bien o para mal.

El código como propiedad

Cuando se habla de los programas para el quehacer editorial, la primera pregunta que surge no es «¿cómo lo uso?», sino «¿cómo lo adquiero?». Las respuestas más comunes involucran su compra o su pirateo. **Tan normal se ha vuelto esta dependencia que se da por sentado que se ha de pagar o violar derechos de autor para poder editar**.

Al final, el código, aunque ya yace ahí en una paquetería de Adobe o de Microsoft, lo hace alguien, le pertenece y tiene la intención de vivir de ello. Por lo tanto, una compensación económica es razonable —o un «discúlpeme, si tuviera dinero le pagaría»—.

La cuestión no es que los programadores tendrían que dar gratis su trabajo, sino la preocupación de quiénes son los que al final le sacan un mayor provecho y para qué.

En el caso de la edición, este desasosiego es qué nuevo programa vendrá a sustituir al anterior, qué tendremos que aprender para no quedar en desventaja, qué habremos de olvidar porque será conocimiento inaplicable; pero, principalmente, ¿cuánto tendremos que desembolsar para pagar por *software*, *hardware* o capacitaciones? ¿Qué de nuevo hay en las ferias del libro que son la panacea *reloaded* de las penurias editoriales?

Cuando no se tiene idea de cómo funcionan las bases de tu profesión es fácil confundir un programa con un método de

trabajo y la dependencia tecnológica con el «futuro» de tu campo.

La falta de estandarización no es solo por la efervescencia propia de la vanguardia, sino también por el interés económico en torno a cómo se realizará el traslado de dependencia tecnológica del grueso editorial. Si se trata del futuro, es el futuro de la presente economía dependiente de la edición al código.

Aunque viendo para atrás nos parezca que la imprenta fue un éxito que cambió al sector editorial, muchas veces se nos olvida que la rotación fue relativamente lenta y que existieron muchas técnicas y tecnologías compitiendo para *abrir* el modo de publicación de los amanuenses.

En esta «batalla» —no tanto porque así haya sido sino por cómo varios amanuenses se sintieron amenazados— también nos falta recordar que la expansión de las tecnologías de la impresión fueron en parte gracias al plagio: todas esas personas que tuvieron acceso al funcionamiento de la maquinaria para replicarla.

El código como bien común

Sin embargo, veo el código de los archivos ejecutables y solo veo unos y ceros. ¿Cómo funciona todo eso?, ¿cómo puedo replicarlo? Y, con mayor interés, ¿cómo hago mi trabajo si desconozco el funcionamiento de la maquinaria que me crea mis libros?

O no nos vayamos tan lejos, veo el texto de una obra ya publicada, la copio para editarla y solo veo contenido. ¿Cuál es su estructura?, ¿cómo fue diseñada? Me quedo con pura información que intento dar formato con mis escasos recursos informáticos para producir una publicación cuya calidad de antemano sé que será inferior al original. Puedo ser usuario y lector de obras

privadas o abiertas, pero mis deseos de ser editor quedan limitados.

Algo tiene que estar mal si para poder aprender a editar tengo que pagar por programas o archivos que no conozco cómo funcionan, y donde mis posibilidades ya están constreñidas por lo que estos ficheros me ofrecen.

De esta manera, surge la idea de que el código no es propiedad privada, sino una cuestión de interés público. El campo de la edición no es el único que ha dado, sin mirar las consecuencias, su soberanía al código propietario. «El código es ley» y si no aprendemos cómo funciona, habrá otros que saquen ventaja de ello.

¿Paranoia? ¿Quiénes fueron los mayores beneficiarios en los traslados de dependencia tecnológica entre PageMaker, QuarkXPress y Adobe, o entre los diversos procesadores de texto hasta Microsoft Office? Mientras tanto, TeX, con ya casi treinta de años, sigue sólido, sin cambios drásticos y con una comunidad pujante que crea libros con alta calidad tipográfica.

En el caso de la edición no estandarizada también tenemos una fuerte competencia entre entornos de desarrollo propietarios y los de código abierto. Las opciones propietarias son más publicitadas y, en varios casos, más sencillas de usar. Sin embargo, con el *software* libre o de código abierto al menos sé que, aunque no sea programador, tengo que tener nociones de cómo funcionan mis programas, y este tipo de *software* me lo permite gracias a su documentación y a que puedo leer el código.

Pero como editor, ¿por dónde empiezo? ¡Tanto conocimiento que se me presenta tan ininteligible!, como fórmulas químicas y matemáticas, cuando mi preparación ha sido principalmente humanística o artística…

La literatura como pedagogía: CLIteratura

¿Y si mostramos la infraestructura informática que está tras bambalinas? ¿Qué tal si hacemos una publicación que a la par que se lee se aprende un poco de los procesos ocultos en la edición? ¿Y si a partir de la literatura aprendemos a usar la computadora de otra manera? Estas preguntas me llevaron a la creación de CLIteratura.

CLIteratura es un pequeño proyecto donde se aprende a utilizar una interfaz de línea de comandos en un entorno emulado en el explorador *web* y mediante la lectura. «CLI» de *command-line interface* y «LIteratura» porque el medio de aprendizaje son los textos literarios.

El proyecto surgió debido a lo que he vivido dando talleres y a lo que he aprendido sobre el modo de hacer libros. Un cambio que considero fundamental en mi vida profesional es haber empezado a **ver los libros como estructuras**, como archivos cuyo formato condiciona ya la calidad editorial y gráfica del producto final.

El giro no fue espontáneo, ha sido un proceso muchas veces tortuoso donde la potencia de la terminal bash no solo me ha ahorrado mucho tiempo, también me ha permitido aumentar la calidad editorial y la técnica de mis publicaciones.

Esto me motivó a crear Pecas: ¿por qué iba a dejar todos esos *scripts* que usaba para mi trabajo y desde bash como una propiedad privada si sé que puede ser, al menos, el primer escalón para que otros se den cuenta de que **la edición es también un método y el texto es nada sin estructura**?

Los profesionales de la edición frente al código

En los tiempos acotados de talleres o pláticas tengo que admitir

que, cuando muestro el modo cómo trabajo, los asistentes quedan más confundidos, y hasta decepcionados de pensar que ellos no podrán tener el mismo control en su trabajo.

Como la mayoría de los eventos a los que soy invitado están orientados a un público versado en humanidades y artes he observado que el uso de muchos ejemplos concretos y de analogías con su campo les permite comprender una serie de nociones que una documentación técnica no podría lograr.

Por estos motivos, durante un tiempo estuve buscando una vía en la que:

- La infraestructura de la información fuera evidenciada, aunque fuese emulada.
- Las humanidades y las artes fungieran como nexo pedagógico, sino que mero pretexto.
- La dinámica fundiera el fomento a la lectura con la enseñanza informática, pese a que bien podría dar un producto híbrido que ni es de aquí ni de allá.

No deseaba que el recurso informático quedara escondido y se diera como dado, sino que se evidenciara y el usuario se familiarizara mientras juega con la herramienta.

Con estas ideas surgió CLIteratura. **Por el momento solo cuenta con un módulo: «Casa tomada», de Julio Cortázar, para aprender a moverse sobre la terminal y a usar las rutas relativas o absolutas**.

Hasta estas fechas solo cuenta con soporte bilingüe inglés-español. Si tu sistema operativo no está en español, de manera automática te cargará la versión en inglés (la traducción de «Casa tomada» es de Paul Blackburn).

Además, quisiera seguir incluyendo más módulos, y sería extraordinario marcar el rumbo de desarrollo mediante su retroalimentación. Úsalo, critícalo o descártalo, todo eso sirve para saber hacia dónde dirigir CLIteratura.

Nota sobre el origen técnico de CLIteratura

La base del emulador de CLIteratura fue desarrollada por Clark DuVall, el cual lo llamó «jsterm».

Lo curioso de jsterm es que fue hecho para su sitio personal, de manera particular como una manera de mostrar su currículo como ingeniero.

¿Cómo se pasó de un sitio para mostrar un currículo a uno para aprender a usar la terminal con obras literarias? Mediante la apertura del código. **Dar acceso al código no solo permite ahondar en la formación del mismo campo de estudio, también posibilita usarlo de maneras inesperadas**.

Sobre esta entrada

- Título original: «CLIteratura, entre el código y la literatura».
- Título en el *blog* de Mariana: «CLIteratura, un proyecto sobre código (de programación) y literatura».
- Fecha de publicación: 31 de julio del 2018.
- Redacción original.
- Revisión de Mariana.

IV. Edición libre

Adobe, el omnipresente

Si se tiene que indicar un elemento tan anclado en la edición mexicana, como el horror ante las viudas y huérfanas, sin titubeos, podemos decir que ese elemento es el uso de InDesign para la maquetación de obras.

Lo mismo cabe decir del ámbito gráfico con Photoshop o Illustrator. Incluso es posible indicarlo para casi toda la producción cultural que se hace a través de una computadora: Adobe está presente en todos lados.

Se parte del quehacer editorial en México, pero considero que se puede generalizar al ecosistema editorial en español, aunque en el campo de las ciencias exactas el paradigma en la publicación sea TeX.

¿Qué hay de malo con Adobe? Nada. La intención de este artículo no va en contra de los productos de Adobe que, en pequeña o gran escala, han mostrado su pertinencia en relación con la relativa corta curva de aprendizaje y los resultados «profesionales» obtenidos. Tampoco se pretende disuadir el abandono de Adobe en pos de otras herramientas (libres) para el trabajo editorial.

De lo que se quisiera hablar es sobre cómo *un entorno de trabajo* ha pasado a ser *el método de trabajo* para la publicación, y los inconvenientes que *pueden* desprenderse de ello.

Adobe como entorno de trabajo

A principios de los ochenta, John Warnock y Charles Geschke

deciden abandonar Xerox para iniciar el desarrollo de PostScript. Siguiendo lo que después sería la tradición de Silicon Valley, desde la cochera de Warnock se fundó su *startup*, la cual llevaría el mismo nombre a un arroyo cercano: Adobe.

Al parecer, en ese lugar se elaboraban ladrillos de adobe para la creciente costa oeste. Siglos más tarde, en ese mismo lugar se empezarían a erigir varias de las grandes empresas tecnológicas impulsoras de la era digital.

El éxito de PostScript fue tal que incluso desde su primer año de fundación Adobe se convirtió en una empresa redituable. Semejantes resultados ayudaron a que PostScript se convirtiera en uno de los primeros estándares para la impresión digital.

Pero no se detuvieron ahí. Con la intención de adentrarse en las oficinas, Apple y Adobe hicieron una alianza. En 1985 el fruto fue cosechado: la primera impresora láser y la primera impresora en red.

La LaserWriter, manufacturada por Apple y con soporte PostScript, fue uno de los pilares para la idea de trabajar documentos de oficina en formato digital o para su posterior impresión.

Poco tiempo después, Adobe decidió crear su propia especificación de fuentes digitales y la nombró **Type 1**. Esto fue el detonante para que Apple creara su propia especificación, la **TrueType**, que posteriormente sería vendida a Microsoft.

La popularidad de la TrueType obligó a Adobe a abrir la especificación de Type 1 para luego llegar a un acuerdo con Microsoft para el desarrollo de una especificación abierta de fuentes: OpenType.

Aunque esto puede entenderse como una batalla perdida, Adobe continuó orientándose al trabajo profesional relacionado con la

impresión. Así es como a finales de los ochenta lanza Illustrator y luego Photoshop.

Con los programas de diseño y el PostScript Adobe constituiría una base para su crecimiento, el cual se fue engrosando con la propuesta de una **«oficina sin papeles»: el PDF**.

El archivo PDF presentaba varias desventajas frente a otros formatos o tecnologías existentes. Por un lado, su gran peso lo hacía poco adecuado para el uso en la incipiente *web* y redes de oficinas. En cuanto uso como documento de impresión, la popularidad de PostScript lo opacaba.

La popularización del PDF fue lenta, pero se consiguió al menos con los siguientes factores:

1. Mejoramiento de la infraestructura del internet, que permitió conexiones a mayores velocidades.
2. Ofrecimiento gratuito de Adobe Reader para el uso y lectura de archivos PDF.
3. Versatilidad al poder ver en la pantalla lo mismo que salía en el papel.

Con PostScript, Illustrator, Photoshop y PDF, Adobe ya estaba abarcando la mayoría del mercado profesional para la impresión. Sin embargo, quedaba una tarea pendiente: el desarrollo de una aplicación para la maquetación digital de documentos, también conocida como *desktop publishing* (DTP).

La incursión de Adobe en ese mercado había sido infructuoso, por lo que a mediados de los noventa decidió adquirir la empresa encargada de desarrollar PageMaker. Cinco años después, en 1999, salió al mercado InDesign.

Casi diez años después, en 2008, la situación era la siguiente:

- InDesign se consolidó como el DTP por excelencia para la publicación digital.
- El PDF se convirtió en un estándar ISO.
- Adobe After Effects ya había sido lanzado al mercado.
- Adobe se había expandido más allá de la esfera profesional centrada en la impresión con la adquisición de Macromedia y, con ello, a Flash, Dreamweaver y Fireworks, entre otros programas.

Había mucho que festejar: **los productos de Adobe ya estaban presentes en cualquier tipo de proyecto cultural**. Y casi otros diez años después, Adobe se ha consolidado como la manera de crear productos culturales, desde la gráfica, pasando por la animación y el video, hasta las publicaciones.

Adobe como método *de facto*

Lo que históricamente se ha constituido como una empresa para ofrecer un entorno de trabajo para casi cualquier tipo de proyecto cultural poco a poco fue mutando en un método.

Entre un entorno y un método hubo un intermediario muy significativo: el aparato publicitario. **Adobe ha promocionado de manera continua sus productos como una solución**, como un entorno para profesionales y creativos de la edición, sea gráfica, multimedia o textual.

¿Cuántos editores profesionales no les agrada la idea de ser asociados con un entorno que da solución a su trabajo del día a día? ¿A cuántos de ellos no les agradan las ideas de «profesionalismo» y «libertad creativa»? ¿Cuántos, al final de cuentas, prefieren una manera fácil de hacer las cosas?

Microsoft Office también es un entorno de trabajo, pero para soluciones «de oficina». Sin embargo, la mayoría de las veces su uso se justifica por lo práctico que resulta optar por formatos omnipresentes como el DOCX, XLSX o PPTX.

En cambio, cuando se habla de Adobe, la practicidad nunca es por una cuestión de formato, tan indiferente es esto que igual se trabaja con formatos abiertos (PNG, PDF, EPS, EPUB, DOCX, etc.) como con privativos (INDD, AI, MOBI y IBOOKS). La practicidad en Adobe reside en la relativa facilidad y libertad posible a través de sus programas para obtener productos con calidad profesional y gran flexibilidad creativa.

Su paradigma no es una oficina, un cubículo o un empleado, pese a que la mayoría de las veces sus usuarios son de esta índole; su estandarte es un espacio abierto, una mesa de trabajo y un profesional independiente. ¿A cuántos diseñadores o editores no nos fascina esta escena?

Esta idea publicitaria necesita un fuerte soporte para ser significativa y efectiva. De la estrategia que ha seguido Adobe para tener presencia en todo proyecto cultural, lo que me parece más interesante es su enfoque visual. Sus programas tienen, del uso básico al intermedio, un **enfoque primordialmente «WYSIWYG»**.

Aunque el término «WYSIWYG» se originó por una manera de tratar el texto digital, donde «lo que ves es lo que obtienes», en el caso de Adobe este enfoque se ha aplicado a casi cualquier esfera de la producción cultural. Lo que se hace en InDesign, Illustrator, Photoshop, etc., es lo que se obtiene en el archivo final o en la impresión.

El trato del contenido digital, no solo para un buen resultado final, sino también para su conservación y distribución, sin lugar a dudas también es una cuestión metodológica: **¿qué pasos tenemos**

que seguir para tener un formato apto para la producción pero que al mismo tiempo se preserve y nos permita llevar a cabo nuestro trabajo?

El *software* ha ayudado a concretar la idea de centralización de cualquier tipo de trabajo a través de una máquina. Sin embargo, como usuarios, este ideal es insuficiente, debido a que resulta molesto brincar de programa a programa, de formato a formato, de codificación a codificación.

Adobe ha permitido la radicalización de ese ideal al ofrecer en un solo entorno, el Adobe Creative Cloud, la solución a proyectos de diversa índole. En un día regular de trabajo quien diseña o edita se las ve con tres entornos:

- el de oficina, lo más seguro que con Microsoft Office;
- el explorador *web*, probablemente Chrome o Safari, quizá Firefox, y
- el de producción, la paquetería de Adobe.

La simplificación y centralización de entornos poco a poco ha impactado en la manera de ejecutar un proyecto, hasta tal punto de que si las tareas no se hacen a través del *software* por *de facto*, surgirán problemas.

No obstante, se espera que el profesional aprenda este método de trabajo. El modo Adobe no solo implica una exigencia al usuario, sino también una normalización del uso de Adobe para incluso otros aspectos del trabajo que no fueron contemplados.

El usuario promedio espera que Adobe ya tenga una solución a sus necesidades y solo sea necesario tomar una capacitación —como sucede cuando se me pide de manera recurrente impartir talleres sobre «libros electrónicos»—.

«Libros electrónicos» entrecomillado porque es común que se piense que la imposibilidad de hacer publicaciones digitales es por una falta de «actualización» o un desconocimiento sobre alguna funcionalidad ya ofrecida por Adobe.

Sin embargo, el carácter problemático de este formato de publicación es de una naturaleza distinta: el problema reside en el modo en cómo se ha estado trabajando con una publicación. La dificultad nace cuando Adobe no es capaz de dar una solución; sin embargo, como instructor el reto yace en demostrar en solo unas cuantas horas que **Adobe no lo es todo**.

Adobe y piratería

El carácter limitado de Adobe para desarrollar publicaciones digitales con calidad es un nicho relativamente muy pequeño en habla hispana, así que ese problema, junto con muchos otros (como la digitalización y el OCR), quedan relegados.

El grueso de los usuarios de Adobe, al menos en el mundo hispanohablante, es para el diseño gráfico, la edición de efectos especiales y el diseño editorial. Y durante años Adobe ha ofrecido y mejorado la manera en cómo se usa su *software* para el cumplimiento de estas tareas. Por este motivo, los problemas relativos a estos quehaceres no son comunes y, de existir, probablemente se deban a errores del usuario.

Esta practicidad y potencia no solo han popularizado los productos de Adobe, también ha atraído algo «indeseable»: la piratería. El uso de *software* pirata de Adobe es tan común como habitual es encontrar que los diseñadores gráficos o editores solo sepan utilizar Adobe para la consecución de sus proyectos.

Esto no es ningún secreto, ni entre profesionales ni entre los

ejecutivos de Adobe. Una de las soluciones a las que se llegaron fue dejar de vender el *software* como programa y, en su lugar, ofrecer una membresía mensual para el uso de toda la paquetería, con descuentos para alumnos y profesores.

En una proyección a largo plazo, el uso de membresías es igual o incluso más redituable que la venta individual de licencias de uso, ya que permite que varios usuarios al fin puedan adquirir productos originales de Adobe. Sin duda es un ganar-ganar.

No obstante, al menos en países en vías de desarrollo como México, aún con esta caída de precios no es posible el uso legal de la paquetería de Adobe. Para muchos diseñadores o editores el pago mensual de 49.99 dólares tiene un impacto directo en sus bolsillos. Por otro lado, el pirateo de *software* es una práctica tan normalizada que incluso se lleva a cabo cuando no existe necesidad.

Al final de cuentas, **la piratería de *software* no es tan indeseable** como se pensaría en un principio. El modo Adobe no se centra en el uso legal o ilegal del *software*, ni la buena salud del entorno de trabajo ofrecido por esta compañía tiene sus bases en la adquisición de licencias de uso.

Lo primordial en el modo Adobe es la prolongación de su método de trabajo. El futuro de Adobe no reside en los programas que desarrolla, sino en la terciarización de sus actividades. El mercado generado por Adobe no solo implica competidores que también busquen desarrollar *software* semejante, sino también toda esa gama de soporte adicional que florece alrededor como los talleres, las certificaciones, las capacitaciones, el soporte técnico, etcétera.

La piratería permite a Adobe la diversificación de sus actividades y el mantenimiento de su estructura como el modo

***de facto* en el diseño**. Si Adobe no ha sido punitivo, como otras empresas de desarrollo de *software*, es debido a que existe una ventaja estratégica en la piratería de su *software*.

Si Adobe ha decidido dejar de vender programas y en su lugar ofrecer membresías para su uso es porque el modelo económico de la cultura de Silicon Valley apunta a uno basado en servicios. Si Adobe tomó la decisión de hacer del PDF un formato abierto se debe a que su omnipresencia le ha permitido tener una cuota de mercado sin necesidad de mantenerlo privativo.

Si en el futuro Adobe ofrece gratis sus aplicaciones será porque es el nuevo camino de la cultura de Silicon Valley (véase a Google, Apple e incluso a Windows, por ejemplo) y a que ya no existirá la necesidad de mantenerlos privados, ya que las estrategias que se han estado tomando le permitirán ofrecer sus productos gratuitamente. Y la piratería ayuda a la consecución de estos objetivos.

Adobe y currículo

La omnipresencia de Adobe no solo es por su enfoque visual, por sus resultados profesionales, por el ofrecimiento de un solo entorno o por un modo de trabajo que se prolonga incluso con su piratería. Su presencia en todos lados también tiene sus bases en dos factores:

1. El conocimiento de su paquetería de programas es un requisito necesario para casi cualquier vacante en el sector cultural.
2. Su aprendizaje no empieza en el «mundo real» del trabajo, sino desde la etapa de preparación profesional en

universidades, institutos, diplomados o talleres.

Lo que Adobe ha provocado es lo que casi toda compañía tecnológica aspira y lo que varias instituciones públicas o privadas desearían: hacer del dominio del *software* una cuestión pedagógica cuya carga curricular se dé durante la etapa de preparación profesional.

Si el uso de la paquetería de Adobe se considera «sencillo» e «intuitivo» en gran medida se debe a que su aprendizaje empieza en la época universitaria. ¿Quién no recuerda el quebrantamiento de cabeza que era entender el uso de cada una de las herramientas de InDesign, Photoshop o Illustrator cuando se tenía una tarea o un examen? ¿Qué tan distinto sería la adopción de TeX en el mundo editorial si su aprendizaje se diera con anticipación, en lugar de lidiar con él cuando así lo exige el trabajo?

Al menos en México, el conocimiento de Adobe se da durante el periodo de educación universitaria o, en todo caso, no es responsabilidad de la empresa o institución que da trabajo al profesional ofrecer ese conocimiento. La competencia pedagógica recae en las instituciones educativas o en el interesado, lo cual en muchos casos no es económicamente accesible para el estudiante.

Los costos requeridos para ello, desde la adquisición de equipo de cómputo adecuado, pasando por el pago de una matrícula o de un taller, hasta *tal vez* el pago de una licencia, no son absorbidos ni promocionados por Adobe. Este grado de penetración que ha tenido Adobe es lo que otras empresas tecnológicas aspiran tener: la necesidad de publicidad, de talleres gratuitos o de demostraciones es prácticamente nula, ya que su *software* se vuelve sinónimo de producción cultural.

Ya no se habla de aprender *desktop publishing*, sino de aprender

InDesign; no se habla de mapa de bits, sino de Photoshop; no son vectores, sino Illustrator, etcétera. La extrapolación entre *software*, formatos y métodos es tal que uno de los retos cuando se demuestra que Adobe no lo es todo para la publicación reside en distinguir entre una y otra cosa, algo tan básico, pero al mismo tiempo tan ignorado por gran parte del sector editorial.

El beneficio que acarrea la disminución de la membresía para alumnos o profesores es claro: **el modo Adobe no solo se sostiene por un entorno suplantado como método de trabajo, sino también por la exigencia curricular por parte de empleadores y educadores**.

¿Pos-Adobe?

¿Y qué tiene de malo? Nada. Entonces, ¿cuál es el problema? La dificultad surge cuando desde Adobe se pretende obtener una calidad de trabajo para aquello que no forma parte de su ecosistema. Las publicaciones digitales han puesto esta carencia en medio del debate; sin embargo, es una cuestión que ha existido desde el mismo surgimiento de Adobe.

Adobe hace uso del enfoque WYSIWYG para todo lo relativo a la producción cultural; es decir, **el modo Adobe tiene su principal enfoque en la dimensión visual** de un proyecto. Si bien para lo que atañe a la edición existe el empleo de estilos de párrafo y de carácter, así como la posibilidad de importar XML, la realidad en el día a día de la edición en México es que los estilos son más una excepción que una regla y la importación es prácticamente desde archivos DOCX.

El enfoque visual no es problemático hasta que existe la necesidad de una estructura consistente. Esta dificultad ha sido

percibida desde sus inicios por dos nichos: la publicación en ciencias exactas y la redacción de documentación de *software*. Así que no importa qué tanto cuidado se tenga con una publicación digital a través de Adobe, su relación costo-tiempo-resultado es menor a otros medios para el desarrollo de esta índole de publicaciones.

Si una publicación digital a partir de InDesign tiene sus carencias o falta de control (donde «control» implica el aprendizaje de tecnologías *web* y la estructura de un EPUB, intención contraria a lo que se pretende con este programa) no es porque el usuario no sea capaz de desarrollar una publicación digital con calidad técnica y control editorial. Se debe a que **InDesign no fue pensado para la publicación digital**, sino para la maquetación de impresos.

En una «era» donde es necesario la publicación multiformato, esto implica que:

- se acepta la escasa calidad de una publicación digital bajo el argumento de la limitación tecnológica o la poca exigencia del lector promedio;
- se implementa un método cíclico de edición que implica un aumento proporcional de costos y tiempos;
- se busca un nuevo método que venga a encarar este reto (véanse los artículos sobre edición cíclica y ramificada, empezando por aquí).

Desde esta perspectiva tenemos las consideraciones metodológicas, técnicas y de cuidado editorial. Pero otro punto es la conservación de los documentos.

Todo proyecto editorial «exitoso» requerirá por lo menos una reedición de la obra. Partiendo del supuesto que la mayoría de las

editoriales o editores solo apuestan por proyectos que serán «exitosos», esto implica que en un par de años se necesitarán de nuevo los últimos archivos editados de una obra.

Debido a las correcciones al vuelo durante la maquetación en InDesign es casi seguro que ese archivo «final» ya solo esté disponible en INDD. ¿Qué pasa cuando la versión de InDesign ya no es compatible con el archivo? ¿Qué pasaría si Adobe descontinúa InDesign, como lo hizo con PageMaker? ¿Qué sucedería si otra empresa entra a la escena del DTP y el usuario empieza a migrar a su ecosistema, como le sucedió a QuarkXPress cuando llegó InDesign?

Los proyectos editoriales con miras a conservarse a un largo plazo no pueden estar a merced de lo que le sucede a las empresas de *software*. Por estos motivos, principalmente en las esferas de influencia anglosajona o germánica, la migración se ha dado hacia archivos XML. El XML también ha sido el formato por defecto para repositorios académicos, principalmente debido a su apertura y su capacidad para el intercambio de información o de publicación automatizada.

Aunque InDesign permite la importación de archivos XML, la imposibilidad de automatización (por el mismo carácter privativo del programa) ha inclinado a varios editores a optar por otras vías, principalmente de la familia TeX, como ConTeXt o TeXML.

¿Que en la automatización se pierde mucho de la rigurosidad ortotipográfica? Cierto, pero a muchas editoriales anglófonas o germáfonas parece no importarles. ¿Debería ser esto relevante también para el habla hispana? Quizá lo discutamos en otro momento.

Mientras tanto, no hay necesidad de buscar un puerto seguro si algún día Adobe deja de ser sinónimo de una profesión, porque las alternativas han estado ahí desde hace mucho tiempo. Solo es

cuestión que el editor promedio se dé cuenta que el control estructural de una obra es primordial y que **Adobe, el omnipresente, no es omnipotente**…

Sobre esta entrada

- Título original: «Adobe, el omnipresente».
- Título en el *blog* de Mariana: «Adobe, el omnipresente».
- Fecha de publicación: 1 de agosto del 2017.
- Redacción original.
- Revisión de Mariana.

> El siguiente artículo supone un conocimiento de lo que es Creative Commons y el *copyleft*. Si no se cuentan con los elementos, se recomienda revisar el sitio de Creative Commons, los tipos de licencias Creative Commons y la entrada en Wikipedia sobre *copyleft*.

Hace algunos años, cuando empezaba a revisar escritos de Lawrence Lessig —impulsor de las licencias Creative Commons (CC)—, encontré una muy buena traducción de *Free Culture*. Esta edición en español tiene una licencia CC y está disponible su versión digital sin costo alguno. Pero es un PDF y yo quería un EPUB. Revisando el sitio de esta editorial encontré casos similares con otros títulos a su disposición con CC y en PDF.

Me surgió una idea muy sencilla: ¿por qué no les pedía los archivos originales *únicamente* para hacer una versión EPUB? Continuaría respetando los términos de la licencia CC BY-NC (usar la obra dando atribución y sin fines comerciales).

Además, si a ellos no les agradaba la idea de que el EPUB estuviera disponible en otros sitios —como en Internet Archive o LibGen— también estaba dispuesto a entregar ese material para que solo en su sitio estuviera disponible, sin necesidad de atribución por su desarrollo.

Con entusiasmo les escribí. ¿Quién negaría trabajo voluntario para tener sus libros en otros formatos? Pasaron algunos días y la respuesta fue clara: no estaban interesados en este momento y, si lo quería, podía hacerlo por mi cuenta a partir del PDF: *no podían entregarme los archivos originales*.

Hacer un EPUB a partir de un PDF no es una tarea difícil, pero es un trabajo innecesario. Cuando partes de un PDF existe la necesidad de volver a darle formato a todo el libro. No es complicado localizar encabezados o bloques de cita.

Sin embargo, la tarea se va incrementando conforme se tiene que dividir el cuerpo del texto de las notas, eliminar guiones de separación, encontrar resaltes tipográficos —itálicas, versalitas o negritas—, sanar el cuerpo bibliográfico y verificar párrafo por párrafo que el contenido esté en su orden correcto, ya que cuando se extrae el texto de un PDF en ciertas ocasiones algunos párrafos aparecen en otro lado de la obra.

Todo esto puede evitarse si en lugar de partir del PDF se usan los archivos que se usaron para maquetar el documento; por lo general, los archivos de InDesign o de algún procesador de texto, e imágenes en PS, JPEG, TIF o PNG.

Aún más tiempo implica si el texto del PDF es imagen con OCR, ya que, por lo general, al menos un 5 % de las palabras fueron mal reconocidas, o si hay imágenes intercaladas en el texto, que también se tienen que extraer y enmendar.

Cuando el autor o el editor no desea entregar estos archivos, implica al menos la necesidad de realizar una lectura de la obra. Si hablamos del caso de la prestación de un servicio, esto implica que el autor o el editor terminarán por pagar más por el desarrollo del EPUB debido a que existe la necesidad de **trabajo adicional redundante** tanto en el aspecto técnico como de cuidado editorial.

La situación con esta traducción de *Free Culture* no es aislada. Con un enfoque exclusivo a obras con *copyleft* o de dominio público, la falta de disponibilidad de los archivos originales, e incluso la indisposición de liberarlos por el temor a la piratería —cuando en realidad las licencias que usan suponen ciertas reglas de uso, sin

importar que sea un PDF o sus archivos de maquetación—, es el pan de cada día.

En este sentido podemos decir que la mayoría de los casos **estos tipos de licencias permiten un uso *abierto* mas no *libre* de las obras**, pese a que se inscriben en el movimiento de «*free culture*» y no de «*open access*».

Lo gratuito, lo abierto y lo libre

Los movimientos de cultura libre y de acceso abierto a la información tienen un antecedente en común que es ajeno a la tradición editorial: el surgimiento del *software* libre y su principal bifurcación, el movimiento del código abierto.

Aquí «tradición editorial» no solo implica una serie de técnicas y de conocimientos editoriales que surgen de tecnologías análogas desde el surgimiento de la imprenta. También comprende las doctrinas jurídicas que desde el siglo XVII fueron constituyéndose en torno al quehacer editorial y que actualmente se conocen como los derechos de autor.

En 1983, ante la masificación de las computadoras personales y la intromisión de grandes corporaciones en el desarrollo de *software* dentro y fuera de las universidades estadounidenses, un grupo de programadores —donde el más conocido es Richard Stallman— deciden hacerle frente a la situación mediante el movimiento del *software* libre, el precursor de la Free Software Foundation (FSF), el proyecto GNU y la licencia GPL. En general, la «libertad» se entiende en cuatro sentidos:

1. libertad de usar el programa (libertad 0).
2. libertad de estudiar y modificar el programa (libertad 1).

3. libertad de distribuir copias del programa (libertad 2).
4. libertad de mejorar el programa (libertad 3).

Para mayor información se recomienda revisar el apartado «Filosofía del Proyecto GNU» del sitio de la FSF.

Estas libertades implican no solo tener el programa ejecutable, sino también el acceso al código. Además, para evitar que posteriores modificaciones del programa queden sin ser accesibles, es menester algún tipo de mecanismo legal que impida que esto suceda.

Así es como nace la idea de las licencias de uso, que si bien en ningún caso es una negación de los derechos de autor o de patentes —al menos no desde un marco jurídico—, sí es una flexibilización en la gestión de lo que ahora se aglutina en el término de «propiedad intelectual». La explicación dada por Creative Commons no podía ser más clara: **no es *todos* sino *algunos* derechos reservados**.

Una de las primeras licencias de uso fue la creada por la FSF: la General Public License (GPL). Una de las principales características de esta licencia es su carácter hereditario: todo lo que se cree a partir de un programa liberado con GPL tendrá que tener a su vez GPL. Desde *El manifiesto de GNU* se explicita que esto se debe a que es una forma de tener garantizada la libertad en los cuatro sentidos mencionados con anterioridad.

Sin embargo, un grupo de programadores que formaban parte del movimiento no les agradó el carácter moralizante de lo entendido por «libertad» y los mecanismos necesarios para defenderla. Lo hereditario se percibió como virulento y, ante el desacuerdo, en 1998 se bifurcó el movimiento del código abierto (*open source*).

Uno de los integrantes más conocidos es Eric Raymond cuyo escrito *La catedral y el bazar* hace patente el desapego de la cuestión moral, así como se dan una serie de «aforismos» prácticos para el desarrollo de *software* en un esquema abierto.

Una de las principales diferencias entre el *software* libre y el código abierto es que el primero explícitamente hace hincapié en el carácter ético y social del desarrollo de *software*, mientras el segundo se aboca más a la cuestión pragmática que implica este desarrollo.

En el *software* libre, ningún programa que forme parte de un proyecto libre puede ser privativo, cuyo autor no ha hecho público el código, normalmente como estrategia para su venta. En el código abierto es posible convivir con ciertos elementos privativos, si en la práctica se quiere conseguir un producto lo más abierto posible o hacer funcionar el *hardware* de un dispositivo.

¿El *software* libre o de código abierto es gratuito? Sí y no. Cualquier persona puede descargar de manera gratuita el *software* libre o de código abierto, pero esto no impide que alguien pueda vender este *software*, sea para recuperar los costos marginales o como forma de conseguir recursos para un colectivo u organización. Por esto **lo gratuito, lo abierto y lo libre no siempre son sinónimos**.

Creación y edición abierta

La discusión dentro del desarrollo de *software* pronto se generalizó al quehacer cultural en general. La pretensión ética y social del *software* libre catapultó sus inquietudes a lo que Lessig empezaría a trabajar bajo el término de «cultura libre».

El enfoque pragmático para el acceso al código permitió la

gestación de lo que ahora conocemos como «acceso abierto» (*open access*). Si bien lo abierto y lo libre en muchas ocasiones conviven sin problemas, en el acceso abierto lo primordial es la disponibilidad del contenido. En este sentido, los colectivos, organizaciones o instituciones enfocadas al acceso abierto no tienen la necesidad de ir más allá de la apertura en el producto cultural *final*.

En la cultura libre se insiste en la relevancia social de la apertura en el acceso de las creaciones culturales. Pero también la cultura libre implica una postura ética hacia nuestra cultura. Creative Commons es una síntesis de estos esfuerzos al ser un conjunto de licencias de uso pensadas para aquellos *autores* que desean permitir la distribución y uso de su obra bajo la confianza de que el usuario no empleará su creación para otros fines a los permitidos por estas licencias.

Creative Commons es una apuesta social y ética no punitiva respecto a nuestra herencia cultural orientada a la *creación*. Y es aquí donde empezamos a ver sus límites cuando en lugar de *creación* hablamos de *edición*. Una edición con CC únicamente tiene la necesidad de hacer accesible el producto *final* de un conjunto de quehaceres culturales.

Por ejemplo, si hablamos de la edición de una obra, hay esfuerzos conjuntos de los autores, los editores, los diseñadores y más personas para producir un libro. Esta edición, si tiene CC, ha de estar a disposición pública para su distribución y uso. Sin embargo, en ningún caso esto implica que los archivos que fueron empleados para producir la obra tengan que ser liberados: solo existe el compromiso de abrir el producto final.

¿Qué inconvenientes tiene esto para la edición? El solo tener el acceso al producto final implica que en muchos casos será

necesario un proceso de ingeniería inversa para dar de nuevo con una serie de contenidos útiles para la reedición de una obra.

La extracción del texto y de las imágenes, y su reformateo o enmendación es ingeniería inversa, que una vez dominada la vuelve un trabajo innecesario. La ingeniería inversa es un buen método de aprendizaje, pero una vez pasado el umbral pedagógico se convierte en una actividad monótona y consumidora de una enorme cantidad de tiempo.

En este sentido, **las licencias cc dentro en el mundo editorial dan fruto a obras abiertas mas no libres**, ediciones cuyo único acceso es el producto final, afín para el usuario y el lector promedio, pero rara vez útil o funcional para editores o desarrolladores interesados en crear nuevas ediciones u otros formatos.

Edición libre

Aunque al hablar de «edición» se hace mención a la «edición de publicaciones», el aspecto editorial cubre otro tipo de productos culturales cuya ingeniería inversa rara vez permite dar con información con la misma calidad que los contenidos originales utilizados para su creación. Los casos más evidentes es la edición de música o de videos, donde la separación de las distintas capas visuales o sonoras es una tarea muy compleja, sino inútil.

Por la falta de necesidad de liberar el contenido originario de un producto final licenciado con cc, surge la idea de la «edición libre». Otro de sus antecedentes pueden encontrarse en la inquietud dentro del *software* libre por crear documentación —como manuales impresos— que al mismo tiempo sean libres; es decir, que no exista la necesidad de pagar por ellos —aunque tampoco se

impide esta posibilidad— y que cualquier usuario pueda reeditarlos.

La edición libre, más que antagónica a la edición abierta, es hacer explícitas las necesidades implicadas en la edición de material mediante la constitución de ciertas obligaciones para autores y editores para permitir el uso directo de los archivos originales.

Así como en el *software* libre la disponibilidad del código es necesaria, **en la edición libre el acceso a los archivos utilizados para la edición no es negociable**. Por supuesto, esto presenta la ventaja o la desventaja de tener un fuerte sentido ético al anunciar que la apertura del producto final no es suficiente: que la liberación de todo el material empleado es una necesidad para otros editores.

La edición libre también tiene implícito un carácter que incluso dentro del *copyleft* es problemático: la **reedición**. Con las ediciones abiertas o en el acceso abierto el usuario tiene la libertad de usar distintos aspectos de estas obras; p. ej., para citarlos o para hacer obras derivadas. No obstante, el límite de esta práctica se vuelve conflictivo cuando no solo hablamos de uso *derivado* de obras, sino la modificación es estas mismas en pos de una mejor edición.

Cuando en un libro o en *paper* como lectores encontramos alguna errata o información incorrecta, lo usual es responderle al autor o al editor para que este rectifique o corrija la información. Rara es la vez que al propio lector se le permite modificar directamente la obra. **La edición libre implica que cualquier usuario en potencia puede modificar, aunque sea ligeramente, una obra**, incluso para tergiversarla…

La tergiversación o el plagio no son problemas resueltos en la edición libre, pero tampoco lo son en las ediciones abiertas, las licencias CC, el acceso abierto e incluso mediante fuertes

restricciones de los derechos de propiedad intelectual como el DRM. **La edición libre solo quiere decir que como editores no tenemos la necesidad de hacer un doble trabajo**.

Licencias para la edición libre

Licencia de Producción de Pares

La Licencia de Producción de Pares (LPP) está basada en la licencia CC BY-NC-SA 3.0, lo cual implica que se permite el uso de la obra licenciada siempre y cuando no se ignore la atribución, no se utilice para fines comerciales y se comparta con el mismo tipo de licencia. Pero en lugar de pertenecer al movimiento de *copyleft*, se inscribe en el **movimiento de *copyfarleft***. Es decir, permite el uso de la obra siempre y cuando sus autores «conserven el valor del producto de su trabajo», lo cual quiere decir que no solo es «no comercial» sino también «no capitalista».

Muchos lectores podría causarles inquietud el carácter «no capitalista» de la LPP. Lo dejaré como asunto pendiente, ya que lo que quiero tratar sobre la LPP es la cuestión de «libertad» de la edición de obras con LPP…

La Licencia de Producción de Pares no obliga explícitamente a liberar todos los contenidos utilizados para la producción de una obra. En sí no es una licencia de edición libre —ni creo que su interés sea este—. Sin embargo, por la manera en cómo ha sido utilizada por colectivos y organizaciones, en varias ocasiones funciona como una licencia de edición libre, ya que no solo está disponible el producto final, sino los archivos que ayudaron para la producción de la obra.

Un ejemplo de proyecto editorial que utiliza la LPP como edición libre es Ediciones La Social. A diferencia de otras editoriales que liberan sus contenidos con licencias CC, este proyecto editorial también facilita los archivos editables de cada una de sus obras, lo que permite su reedición o creación de otros formatos.

Un ejemplo que va al caso con este tema es el libro *Muestra Libre. Reflexiones sobre Piratería, Software Libre y Diseño*, que está disponible en PDF o en el formato abierto de procesador de textos (ODT). El acceso al archivo editable me permitió desarrollar sin dificultades, junto con una amiga, su versión EPUB, cuya publicación no causó molestia a Ediciones La Social.

Esto no es un caso aislado, sino una muestra de la tendencia de proyectos editoriales que utilizan la LPP en sus obras. Lo que en estos colectivos u organización se hace patente es la búsqueda de crear redes cooperativas de trabajo o al menos de no impedir o interferir en el trabajo realizado por otros.

Mucho ayuda el que no estorba, y de este modo la LPP funciona como licencia de edición libre al no dificultar el quehacer editorial mediante la reserva de los archivos editables. **De la intención de desarrollar un EPUB con la edición en español de *Free Culture* a la creación del EPUB de *Muestra Libre* existe un mar de diferencias que ejemplifica la distinción entre la edición abierta y la edición libre**.

Licencia Editorial Abierta y Libre

Hace algún tiempo en Perro Triste se lanzó una propuesta de licencia de uso en pos de la edición libre: la Licencia Editorial Abierta y Libre (LEAL, acrónimo sugerido por un «hackeditor», porque en un principio fue LELA). **Algunas de las características de**

la LEAL son las siguientes:

- Los autores o editores tienen la obligación de hacer pública los archivos editables utilizados durante la producción de una obra.
- La comercialización es posible, siempre y cuando no sea el único medio de adquisición de los archivos editables o del producto final.
- La licencia supone un acuerdo previo entre las personas involucradas en la producción de una obra para que esta sea abierta y libre.
- La atribución no es necesaria.

La LEAL hace posible que cualquier usuario tenga acceso a los archivos editables, con lo cual se evita la ingeniería inversa al mismo tiempo que fomenta la reedición de las obras sin la necesidad de pedir permiso a los autores o editores «originales».

Ejemplos de uso de la LEAL están la misma LEAL, las publicaciones hechas por Perro Triste, algunos libros elaborados por una compañera de trabajo y el material que se publicó como la «Historia de la edición digital», «Derecho de autor cero», mis entradas en este *blog* y hasta escritos aún en proceso de redacción como «Cryptodoc».

Al permitirse la comercialización, la LEAL no forma parte del movimiento de *copyfarleft* como la LPP. Más que un fomento al uso «capitalista» de las obras licenciadas con LEAL, obedece a la intención ya presente en el *software* libre de no imposibilitar la comercialización al poder ser un mecanismo de apoyo a los proyectos editoriales que publican con LEAL. No obstante, es una de las características potencialmente problemáticas que aún no se ha

analizado a fondo, ya que en sí el *copyfarleft* no prohíbe la comercialización con el fin de mantener vivo un proyecto cultural.

La liberación de los archivos editables muchas veces implica la publicación del trabajo realizado por un grupo de personas, las cuales no en pocas ocasiones estarán de acuerdo en ello. Por esto, la LEAL supone un acuerdo previo, para así evitar que los usuarios que se valen de una obra con LEAL tengan que pedir permisos de uso a cada uno de los involucrados o, peor aún, que se vean implicados en un proceso legal.

El acuerdo previo no es resuelto por la LEAL, sino que este puede ser desde un acuerdo informal o verbal, hasta la firma de contratos o convenios de cada uno de los colaboradores. En este sentido el marco legal presupuesto no está contemplado por la LEAL, lo que también la constituye como un punto débil.

Por último, la ausencia de atribución es contraria a uno de los supuestos dentro del *copyleft* y del *copyright*: en uno u otro caso supone que cada obra *derivada* ha de hacer mención a los autores y editores de la obra *originaria*. Si bien una de las dos tipos de licencias propuestas, la LEAL-A, añade la necesidad de atribución, la LEAL estándar prescinde de ello. ¿Por qué? La LEAL no solo tiene la intención de ser una licencia de uso para ediciones libres, sino abrir la discusión a lo que suponen los conceptos de «atribución», «autoría» y «obra» (*derivada*). Este es el principal punto conflictivo que le he visto a la LEAL y que, por supuesto, requiere discusión.

Como sea, la LEAL es solo una propuesta por estas y otras dificultades —por ejemplo, su relación con otras licencias de uso que solo se enfocan a la creación, como las CC—. Lo relevante aquí es rescatar que **la edición libre va más allá de las licencias Creative Commons, ya que su objetivo no es solo la apertura del producto final, sino explícitamente hacer públicos los archivos editables**.

Si gran parte del sector editorial continúa siendo escéptico a Creative Commons, ¿qué tanta pertinencia tiene un movimiento —la edición libre— que también pretende la publicación de los archivos editables de una edición?

Sobre esta entrada

- Título original: «Edición libre, más allá de Creative Commons».
- Título en el *blog* de Mariana: «Edición libre, más allá de Creative Commons».
- Fecha de publicación: 22 de noviembre del 2017.
- Redacción original.
- Revisión de Mariana.

Lo gratuito, lo abierto y lo libre

En el quehacer cultural —como ya sucedió en el desarrollo de *software*— es cada vez más común escuchar sobre el acceso abierto y la cultura libre. Pero **¿qué es eso de lo «abierto» y de lo «libre»?**

«Bájalo, es gratis»

Uno de los primeros acercamientos hacia los productos o servicios abiertos o libres es que son «gratuitos». El usuario, de repente, se percata que no hay nada que lo restrinja en la descarga o uso de algún programa o contenido, llegando incluso al acaparamiento digital (*e-hoarding*).

La gratuidad puede ser por infracción de algún tipo de propiedad intelectual —como los derechos de autor o de patentes— aunque también se da porque el autor decidió liberar el contenido de esa manera.

Alphabet Inc. es un buen ejemplo de una compañía que ofrece a sus usuarios varias aplicaciones de manera gratuita; por ejemplo, Chrome, Gmail y el motor de búsqueda Google.

Facebook es otro ejemplo, así como Twitter, Airbnb, Uber, etcétera. Para el caso editorial, tenemos el catálogo mexicano de libros de texto gratuito, los diversos diccionarios de la RAE o las consultas en Fundéu BBVA.

Aunque no sea posible generalizar, en la mayoría de los casos **el usuario termina dependiendo de varios de estos servicios «gratuitos»**: el explorador para navegar por internet; el gestor de

correo para el trabajo; el motor de búsqueda para encontrar información; las redes sociales para estar en contacto con familiares, amigos o compañeros de trabajo; los servicios de geolocalización para trasladarse, etcétera.

Algunos usuarios muestran su preocupación por este carácter de dependencia, ya que, más allá de lo que se ofrece, no es posible saber qué tanto se están utilizando nuestros datos personales para otros fines —lo que se conoce como minería de datos— o qué medidas de seguridad se están llevando a cabo para su protección.

El **caso de Cambridge Analytica y Facebook** son los más recientes en este aspecto, pero no son el único, como puede verse en las diversas demandas hechas a Alphabet Inc. por acaparamiento de datos del usuario o a Uber por su fuga de información.

En este sentido puede decirse que **lo gratuito no es siempre sinónimo de apertura o libertad**, ya que por lo general el interés detrás de varios productos o servicios gratuitos es la obtención de datos del usuario con procedimientos poco claros y sin que este tenga mucho margen de maniobra.

Abriendo caminos

Aunque la privacidad digital es un debate público, este tema ha estado desde los inicios de la «era» digital. Durante los noventa y el surgimiento de las empresas puntocom varias personas señalaron los riesgos que el usuario podía tener por compartir su información.

Por un lado, hay compañías que controlan el flujo de la información, muchas veces de manera innecesaria. Por el otro, se cuenta con una infraestructura tecnológica para satisfacer ese fin

que sobresale por su falta de claridad en su funcionamiento, al mismo tiempo que está fuertemente protegida para evitar que esta sea examinada.

La **iniciativa del código abierto** surgió para contrarrestar este problema, ya que apuesta por:

- Abrir el código para que cualquiera pueda examinarlo y mejorarlo, posibilitando así la rápida identificación de problemas de seguridad.
- Disminuir las barreras legales que muchas veces impiden crear o mejorar tecnologías.

Esto implica un cambio en las metodologías de producción de *software*, pero también de las formas de organización y las posturas en torno a cómo se percibe la construcción del conocimiento.

Por estos motivos, a finales de los noventa, la iniciativa del código abierto fue expandiéndose hacia otros ámbitos de la producción cultural, y así surgieron los **movimientos de la «cultura libre» y el «acceso abierto»**.

Con el código abierto los usuarios tienen la seguridad de que un producto o servicio, muchas veces gratuito, no tiene nada oculto en detrimento del usuario. Como ejemplos tenemos a Firefox, Thunderbird, DuckDuckGo, Telegram u OpenStreetMap.

Para el caso del acceso abierto y la cultura libre los diversos productos ofrecidos dejan claro que el usuario no está violando ningún tipo de propiedad intelectual —siempre y cuando se respete la licencia de uso—, así como se le ofrece, «tal cual».

Es decir, el usuario no está condicionado a dar su información personal a cambio del producto; sin embargo, el autor no tiene la obligación de ofrecerle «servicio al cliente». Ejemplos de esto son

Internet Archive o SciELO que, a su vez, protegen su información con algún tipo de licencia Creative Commons.

Y aunque **la apertura deja claro que la prioridad no es tanto la gratuidad de lo que se ofrece, sino el acceso y la privacidad de quien lo usa**, en algunas ocasiones esto es insuficiente. ¿Qué tal si como usuario o autor tu interés no es solo el acceso o la privacidad, sino también la «libertad»?

> *Lecturas recomendadas*: «Vender vino sin botellas» de John Perry Barlow, cofundador de la Electronic Frontier Foundation, y «La Catedral y el Bazar» de Eric S. Raymond, «padre» de la iniciativa del código abierto.

La ambigua libertad

Mucho antes de la iniciativa del código abierto surgió el movimiento del *software* libre —de hecho, el código abierto es una bifurcación de la comunidad del *software* libre—, que percibe a la libertad desde cuatro puntos:

1. Libertad de ejecutar un programa.
2. Libertad de estudiar y modificar un programa.
3. Libertad de redistribuir un programa.
4. Libertad de mejorar y publicar las mejoras de un programa.

Para asegurar que nunca se rompan algunas de estas libertades, el *software* libre viene acompañado de licencias que garantizan que, sin importar el rumbo que tome, todas las versiones tendrán que ser compartidas de la misma manera.

Una de las ventajas que se obtiene con esto es que **nadie tiene la necesidad de pedir permiso para ejercer alguna de estas libertades**, así como el autor cuenta con la seguridad de que su

producto siempre permanecerá libre.

Una de las principales críticas que el movimiento del *software* libre ha hecho al código abierto es que, en algunos de los casos, sus licencias de uso son demasiado restrictivas, ya que impiden la modificación o la redistribución de un programa. Además, también ha criticado el «miedo» hacia el uso de la palabra «libertad».

Por otro lado, algunas personas dentro de la iniciativa del código abierto critican al *software* libre por crear licencias «víricas»; es decir, licencias que *impiden la libertad* de compartir un programa con una licencia distinta, creando así una paradoja. Además, también se ha argumentado el fuerte carácter normativo y la ambigüedad en el uso del término «libertad».

Aunque **en la práctica lo abierto casi siempre es libre y viceversa, existen diferencias con un trasfondo político y filosófico** que se hacen perceptibles en esta oposición.

En la iniciativa del código abierto casi nada interesa más allá de la esfera de organización y el acceso a la información: se enfoca en el modo de producción y su transparencia.

Mientras tanto, el movimiento del *software* libre pretende ser más integral y situar al desarrollo del *software* en un contexto cultural y social más amplio, incluso desde una perspectiva de «ética kantiana» —guiño al imperativo categórico—: obra de tal manera que se convierta en ley universal.

«Si nada de eso me interesa, ¿qué ventaja tiene el *software* libre?»

De manera general, el *software* libre evidencia que el acceso a la información y la privacidad del usuario no es reducible al modo en cómo se desarrolló la tecnología, sino que también abarca los supuestos básicos que sirven de guía para este desarrollo y cómo

estos nos afectan como comunidad o como cultura.

En el caso del quehacer cultural, **la libertad en el uso, modificación o mejora permite un acceso irrestricto y seguro a los contenidos**. No es lo mismo tener acceso al PDF, al MP4 o al MP3 para hacer una reedición o *remix*, que a cada uno de los archivos editables.

Tampoco es lo mismo decir que tu producto es abierto, al mismo tiempo que restringes ciertos usos, que decir que es libre. Ni es lo mismo abrir tu material por moda o porque es más redituable, que liberarlo por el afán —sino que mera necedad— de percibir el valor no monetario de la cultura.

¿Ambigüedad? Sin duda. ¿Utopía? Tal vez. Aún falta mucho por discutir sobre lo libre y más cuando en el sector editorial se empieza a ver en el horizonte no el acceso abierto, sino la **«edición continua y libre»**…

> *Lecturas recomendadas*: «El manifiesto de GNU» y «Por qué el "código abierto" pierde de vista lo esencial del software libre» de la Free Software Foundation.

Sobre esta entrada

- Título original: «Lo gratuito, lo abierto y lo libre».
- Título en el *blog* de Mariana: «Lo gratuito, lo abierto y lo libre (o la gratuidad, el acceso abierto y la cultura libre)».
- Fecha de publicación: 27 de marzo del 2018.
- Redacción original.
- Revisión de Mariana.

Cómo difunde su obra un autor: *copyright*, *copyleft* y *copyfarleft*

Junto con lo gratuito, lo abierto y lo libre acontece una serie de replanteamientos sobre el respaldo legal de cada autor en relación con el modo en cómo difunde su obra. En varios de los casos esto se traduce a una serie de confusiones que vale la pena abordar, aunque sea de una manera muy superficial.

El nacimiento del *copyright* o derechos de copia

El «derecho de copia» o también conocido como *copyright* es una doctrina jurídica que protege al autor y a su obra. Su símbolo es mundialmente conocido (©) y su legislación parece extenderse incluso hasta en contra de la voluntad del mismo autor.

Sin embargo, su origen es un tanto distinto. Con la expansión de la imprenta en Gran Bretaña, algunos autores empezaron a cuestionar el monopolio que tenían varios editores/impresores —por comodidad se usará cada término indistintamente—. Con anterioridad, no existía una legislación clara respecto a la relación entre ellos por lo que en 1710 la Reina Ana promulgó un estatuto que daba plenos derechos a los autores de explotar sus obras.

Este estatuto les daba «derechos de copia» a los impresores, haciendo explícito que no eran los propietarios de la obra. **Este derecho de explotación exclusiva de los editores solo era por un tiempo limitado**. Una vez caducado el lapso, el autor era libre de dar prórroga al contrato o buscarse un nuevo impresor. En caso de fallecimiento del autor, una vez cumplido el lapso, la obra pasaba al

dominio público. **El lapso inicial era de tan solo catorce años** y durante un tiempo permitió una gran flexibilidad y expansión de la industria editorial. Pero también dio hincapié a nuevos conflictos: por un lado, el interés del editor en ampliar el plazo de sus derechos de explotación; por el otro, la preocupación de autores y familiares de no poder lucrar más con la obra una vez fallecido el autor.

Ante estos conflictos, poco a poco las leyes de *copyright* se fueron modificando para incluir la posibilidad de herederos a las obras y así continuar su explotación comercial, o la ampliación de los plazos de explotación.

En el panorama actual, el titular de los derechos puede ser cualquier persona o institución —pública o privada— que el autor señale y sus derechos de explotación pueden extenderse a por lo menos cincuenta años después de la muerte del autor, aunque la mayoría de los Estados es ya de setenta años —como el caso de la Unión Europea y Estados Unidos— o incluso de cien años —como sucede en México—.

Si bien esto ha permitido ampliar los plazos para la adquisición de regalías, también ha generado el problema de las obras huérfanas —trabajos que aún tienen *copyright*, pero cuyo titular de derechos no puede ser localizado, por lo que la obra en teoría no puede ser utilizada—, así como la imposibilidad de emplear un material sin la autorización del titular —lo que supone que cualquier uso no autorizado, aunque sea de elogio, está infringiendo con los derechos de autor—.

El surgimiento del derecho de autor frente al *copyright*

El «derecho de copia» surge en Gran Bretaña al comienzo de la Ilustración, pero como es evidente, no toda legislación moderna

viene del mundo anglosajón. Así como John Locke es uno de los «padres» de esta tradición jurídica, en la Europa continental surge otra tradición: el **derecho de autor**.

Los «padres» del derecho de autor son Kant, Hegel y varios juristas franceses. Este derecho tiene la similitud de que surge para proteger al autor y su obra, pero con la diferencia de que en realidad de trata de dos tipos de derechos: los patrimoniales y los morales. **Los derechos patrimoniales en sus efectos son iguales al *copyright***, son los derechos que permiten transmitir las posibilidades de explotación de una obra. Esto trae implícito que la obra se considera un «fruto del trabajo» del autor, el cual puede decidir la manera más oportuna para sacar «beneficios por su labor».

Sin embargo, autores como Kant y Hegel consideraban que la obra era más que eso: creían una creación era una expresión de la personalidad o la voluntad del creador. La relación entre el autor y su obra no es «accidental» —el trabajo no es sustancial al autor—, sino intrínseca y única —la relación entre el autor y *su* obra es cualitativamente distinta a cualquier relación que esta obra pudiera tener con otra persona—. Es decir, al momento de producir una obra, no solo se *crea un producto* mediante el trabajo, sino que también acontece una intimidad por la cual es posible identificar que **una obra *forma parte* del autor**.

Los derechos patrimoniales se quedan cortos al proteger a la obra desde esta perspectiva, ya que solo la tratan como un producto. Por ello, **los derechos morales funcionan para reconocer la *paternidad* de una obra y el derecho que tiene el autor a proteger su integridad**. Dado a que los derechos morales tratan a la obra como extensión del autor, estos son de carácter intransferible e inalienables —o eso se supone—.

Con esto, los derechos de autor son muy similares al *copyright*, por lo que acarrean las mismas problemáticas que plantea el «derecho de copia». Esto también permite que sea común utilizar el símbolo «©» más la añadidura de un «D. R.», explicitando que los «derechos reservados» engloban tanto los derechos patrimoniales y morales.

Al mismo tiempo, abre paso a ciertas ambigüedades ya que más allá de un ámbito romántico o «metafísico», la supuesta relación intrínseca e íntima entre el autor y su obra no puede explicarse del todo, pese a que en un primer momento parezca evidente.

El movimiento del *copyleft* y el de la cultura libre

Hay varias personas que por motivos económicos, políticos o sociales no están de acuerdo con las legislaciones actuales en torno al derecho de autor. Uno de los principales argumentos es que las leyes actuales restringen el flujo creativo que durante mucho tiempo se dio en el quehacer cultural en pos de casos minoritarios de infracción.

Como propuesta a esta situación, surge el movimiento del *copyleft*, cuyo nombre es un juego de palabras por el que se manifiesta que es algo opuesto al *copyright*. Su símbolo —cada vez más popular— es la inversión del símbolo del *copyright*. Al no existir aún ese carácter en la mayoría de las fuentes tipográficas, como alternativa en texto plano se utiliza una «c» invertida y entre paréntesis: «(ɔ)».

El origen del *copyleft* se remonta a los años ochenta del milenio pasado en el ámbito del *software*. Richard Stallman y el movimiento del *software* libre fueron los primeros en buscar alternativas *jurídicas* que protegieran el código para asegurar su libre

distribución. **Por primera vez en la historia, los autores empezaron a preocuparse por tener un respaldo legal que les permitiera distribuir su obra con libertad**. Es así como surgió la Licencia Pública General GNU o GPL, por su acrónimo en inglés.

A finales de los noventa, estas ideas en torno a la propiedad del código y del *software* se expandieron a los productos culturales en general. Su principal representante es Lawrence Lessig y el **movimiento de la cultura libre**. Las licencias Creative Commons surgen a partir de este movimiento hasta el punto que en la actualidad son las licencias *copyleft* más populares.

La obra como un bien social antes que individual

En general, lo que estas licencias promulgan es un paso del «todo» a «algunos» derechos reservados. Pero más relevante aún, **de manera implícita da a entender que la obra es tanto un producto individual como social**: **un bien común**.

El objetivo es simplificar el aparato jurídico y burocrático para que cualquier persona —según los términos de cada licencia— puedan utilizar un producto cultural como «suyo». La meta es constituir un ecosistema cultural por el cual usuarios y creadores no se vean limitados, al mismo tiempo que asegura la integridad de la creación y su creador. Estos ideales poco a poco han tenido mayor recepción pero también han provocado algunos conflictos. Por un lado, se consideran una amenaza directa a la forma en cómo se ha gestado la cultura desde la Ilustración —aunque en realidad solo ataca de manera directa los intereses monopólicos de algunas compañías o asociaciones, tal como el *copyright* surgió para eliminar el monopolio de editores e impresores—. Por el otro, crea un nuevo ecosistema cultural el cual se considera que debería de

ser más regulado por los Estados.

El *copyleft* y la interpretación del derecho de autor

El problema de la regulación afecta el ámbito legislativo ya que ninguna tradición jurídica —anglosajona o continental— había considerado la posibilidad de que el autor quisiera proteger «libremente» su obra. De hecho, la «libertad» del autor se había comprendido como el respecto y protección de su obra como su «propiedad» y las «libertades» económicas y de comercio que de esto se desprendía.

Esta otra noción de lo «libre» —donde la obra es un «bien común» y no una «propiedad»— y la demanda de los ciudadanos en su respaldo jurídico ha inducido a la modificación de las distintas legislaciones. En sentido jurídico, **las propuestas de licenciamiento del *copyleft* no son una sustitución del derecho de autor, sino un «suplemento»** por el cual se constituyen como licencias de uso no exclusivas.

Es decir, los derechos de explotación exclusiva de una obra no son transferidos ni la obra pasa a un dominio público voluntario, sino que siguen perteneciendo al autor o sus herederos. Esto no sería conflictivo si de manera efectiva se respetara la voluntad del autor por dejar «abierta» su obra aun después de su muerte, pero no siempre se da el caso, como ya se mencionó en «Derecho de autor cero». Esto hace que, pese a las intenciones del movimiento del *copyleft*, en la actualidad distintas legislaciones interpreten sus licencias como una extensión a los derechos de autor. Idea que parece un tanto tergiversada, aunque quizá no lo sea tanto.

Existe el derecho de autor y el carácter opcional de permitir el uso «libre» de su obra. Por tanto, ¿qué problema hay en todo esto?

Para algunos críticos, esta «apertura» en el uso de la obra no toma en cuenta factores políticos y económicos que afectan directamente el bienestar de un autor: ¿qué pasa si una obra con *copyleft* es explotada de tal manera que un tercero obtiene un beneficio económico que no es compartido con el autor?

Supongamos que una artista *X* decide publicar su obra *Y* con licencia CC-BY —solo pide atribución a la obra—. A continuación, una empresa *Z* ve un gran potencial comercial de *Y*, por lo que comienza con su explotación. Por último, resulta que esta explotación ha dado grandes beneficios económicos a *Z* y, dado al tipo de licencia, esta no está comprometida a pagarle regalías a *X*, acumulando así todos los beneficios.

Las palabras clave aquí son «explotación» y «acumulación»: para estos críticos si bien **el *copyleft*** pretende intrínsecamente otro medio de compartir y crear cultura, **no implica la eliminación de la explotación de los artistas por parte de un sistema económico de acumulación de capital**.

Es decir, **el *copyleft*** se queda corto al no querer ir más allá de la manera en cómo se relacionan autores y consumidores. Peor aún, **prolonga la idea capitalista del «libre mercado»** al buscar disminuir o simplificar las funciones tradicionales de los Estados —como la de ser un regulador entre el autor y su público o las entidades que explotan su obra—.

Esto genera un problema ético de fondo: ¿*debe* la compañía *Z* repartir el capital acumulado con el artista *X*? Jurídicamente no tiene el *deber* de hacerlo, pero desde un enfoque ético *debería*

llevarlo a cabo, de lo contrario, estaría prolongando la cadena de explotación del «obrero/autor» por parte del «capitalista».

¿Qué se podría pensar si la artista *x* no tiene otro medio de subsistencia y, al mismo tiempo, la compañía *z* está lucrando de manera constante con su obra *Y*? ¿Acaso no es la prolongación de una situación injusta? Para evitar este problema, surge la iniciativa del *copyfarleft*, la cual es también un juego de palabras que implica ir aún más a la «izquierda» del espectro político del *copyleft*. La principal característica de los tipos de licencia propuestos por esta iniciativa es la adición del requisito «no-capitalista».

La Licencia de Producción de Pares sintetiza muy bien esta singularidad:

> No Capitalista - La explotación comercial de esta obra solo está permitida a cooperativas, organizaciones y colectivos sin fines de lucro, a organizaciones de trabajadores autogestionados, y donde no existan relaciones de explotación. Todo excedente o plusvalía obtenidos por el ejercicio de los derechos concedidos por esta Licencia sobre la Obra deben ser distribuidos por y entre los trabajadores.

¿Qué problema puede tener este tipo de licencias? En un comienzo los prejuicios hacia su propio discurso: son pocas las personas que tienen empatía ante críticas socialistas, marxistas o anarquistas, más aún en un ecosistema cultural definido a partir de subsidios estatales, «empresas culturales» o el «emprendimiento individual».

El *copyfarleft* y el derecho de autor

Como licencia de uso, el *copyfarleft* es tratado como un tipo de licencia *copyleft*. Sin embargo, el apego a su criterio no-capitalista puede no ser claro para la mayoría de las legislaciones.

¿Qué se entiende por «sin fines de lucro»: cualquier tipo de colectivo o solo a las organizaciones no gubernamentales sin fines

de lucro debidamente registradas? ¿Qué se entiende por «autogestión»? ¿Qué se interpreta por «explotación»?, etcétera…

No existe certeza jurídica sobre el respeto a este tipo de licencias; sin embargo, el meollo no es de índole legislativa, sino política. **El *copyfarleft* tiene el interés de hacer explícito el espectro político del autor y su obra**. Y **el seguimiento a su principio no-capitalista es, ante todo, un acto ético y de respeto a estos principios**. Y tú, ¿qué tanto te vas a la «derecha» o a la «izquierda»? ¿Son estas tres tendencias hacia los derechos de autor completamente disímiles?

Sobre esta entrada

- Título original: «*Copyright*, *copyleft* y *copyfarleft*».
- Título en el *blog* de Mariana: «Cómo difunde su obra un autor: *copyright*, *copyleft* y *copyfarleft*».
- Fecha de publicación: 20 de abril del 2018.
- Redacción original.
- Revisión de Mariana.

Anexos

«Historia de la edición digital» es la segunda entrega sobre edición digital, de **Ramiro Santa Ana Anguiano**, de **Nieve de Chamoy**. Debido a la extensión, más larga que otras entradas, hemos preferido publicar este artículo en archivos descargables.

Los formatos disponibles en los que se encuentra el artículo son EPUB, MOBI, GitBook y PDF. Gracias a esta variedad, puedes elegir el formato que te resulte más cómodo para leer.

También se pueden **descargar los archivos editables** para apreciar los métodos descritos por Ramiro en este artículo.

Los cuatros archivos fueron creados por Ramiro desde la metodología propuesta en el artículo; a saber, *single source publishing* junto con *online publishing*.

Todos los formatos se encuentran alojados en GitHub, también el registro del proceso de edición.

A partir de un Markdown, como archivo «madre», se crearon los siguientes formatos:

- **EPUB** desde cero y mediante los *scripts* de Perro Triste. **Clic para descargar**.
- **MOBI**, desde EPUB, con la herramienta de conversión de Amazon: KindleGen. **Clic para descargar**.
- **PDF**, mediante Pandoc y LaTeX. **Clic para descargar.**
- **GitBook** desde cero (donde solo se divide el documento de Markdown en las secciones correspondientes ya que los archivos base de GitBook ya son Markdown). **Clic para**

visualizar.

Temas que aborda «Historia de la edición digital»

A modo de resumen, a continuación incluimos un punteo de los temas que Ramiro desarrolla en este artículo:

- Cómo se introdujeron las computadoras u ordenadores a la cotidianidad del público general.
- Qué son los lenguajes de marcado y el lenguaje de marcas ligero.
- Por qué la industria editorial no adoptó TeX, un programa pensado para el cuidado tipográfico y composición de textos.
- Qué son los enfoques relacionados al tratamiento del texto digital WYSIWYG y WYSIWYM, y cómo afectan a la edición digital.
- Cómo y por qué surgió la *desktop publishing* (DTP).
- La ideas entorno a la «libertad» en la edición que plantean el *software* privativo y el *software* libre.
- El surgimiento de la *web* y la consiguiente aparición del HTML, el XML y el XHTML.
- La aparición del EPUB y los estándares W3C.
- Diferencia entre *edición digital* y *publicación digital*.
- Propuesta para la automatización y sistematización de la edición digital
- Qué son *online publishing* y *simple source publishing* y qué relación poseen con el control sobre la edición y las versiones de archivos.

Palabras clave: Adobe Creative Suite, Amazon, Apple, código

fuente, CSS, cuidado editorial, *desktop publishing* (DTP), *ebook*, EPUB, formateo, formato, Gimp, GNU, *hardware*, HTML, IBM, Illustrator, InDesign, Inkscape, LaTeX, lenguaje de marcado, LibreOffice, libro impreso, Markdown, MOBI, OpenOffice, PDF, Photoshop, PostScript, proyecto Gutenberg, publicación digital, QuarkXPress, repositorios, Scribus, *simple source publishing*, *software*, *software* libre, TeX, W3C, Word, WordPerfect, WYSIWYG, WYSIWYM, XHTML, XML.

A diferencia del resto de los contenidos de este blog, **«Historia de la edición digital» posee Licencia Editorial Abierta y Libre (LEAL)**.

Sobre esta entrada

- Título original: «Historia de la edición digital».
- Título en el *blog* de Mariana: «Historia de la edición digital».
- Fecha de publicación: 10 de octubre del 2016.
- Repositorio.
- Revisión de Mariana.

Tras bambalinas de «Historia de la edición digital»

Hace unos meses atrás publicamos el artículo **«Historia de la edición digital» en cuatro formatos digitales diferentes: EPUB, MOBI, PDF y GitBook**. Estos formatos se crearon según la metodología del *single source* y *online publishing*, abordada en esta entrada.

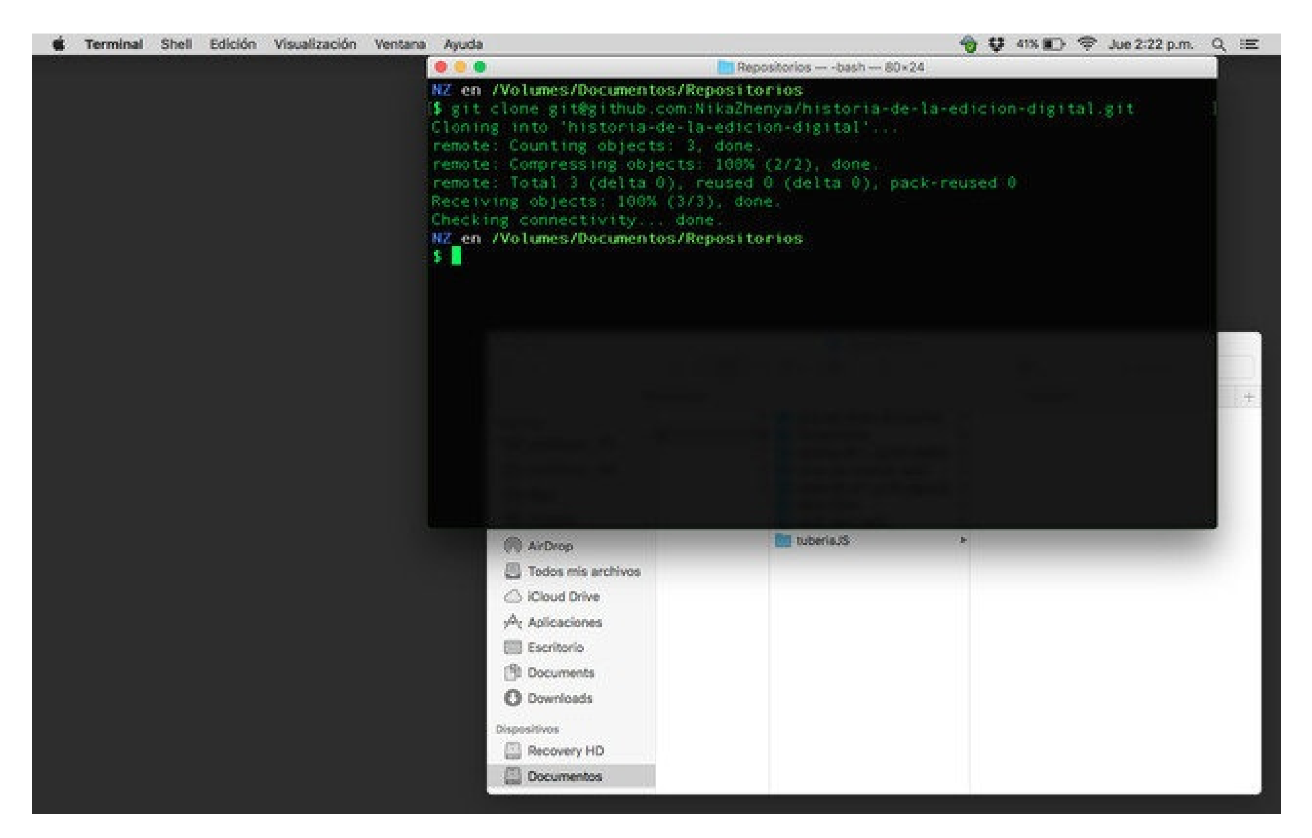
Aunque te cueste creerlo, los cuatros formatos fueron procesados y creados en un día, en el tiempo de una jornada de trabajo.

Como anticipamos en el repositorio del artículo, en esta entrada se explican los pasos llevados a cabo para su edición y publicación.

Proceso de desarrollo

A continuación se comentarán unas impresiones de pantalla omitiendo la parte técnica, ya que los *scripts* utilizados están pasando por grandes cambios en aras de la posibilidad de una interfaz gráfica. Estas son las herramientas, por si quieres echarle un ojo.

1. Clonación del repositorio, 14:22 h



Clonación del repositorio, 14:22 h.

El proyecto es un repositorio para:

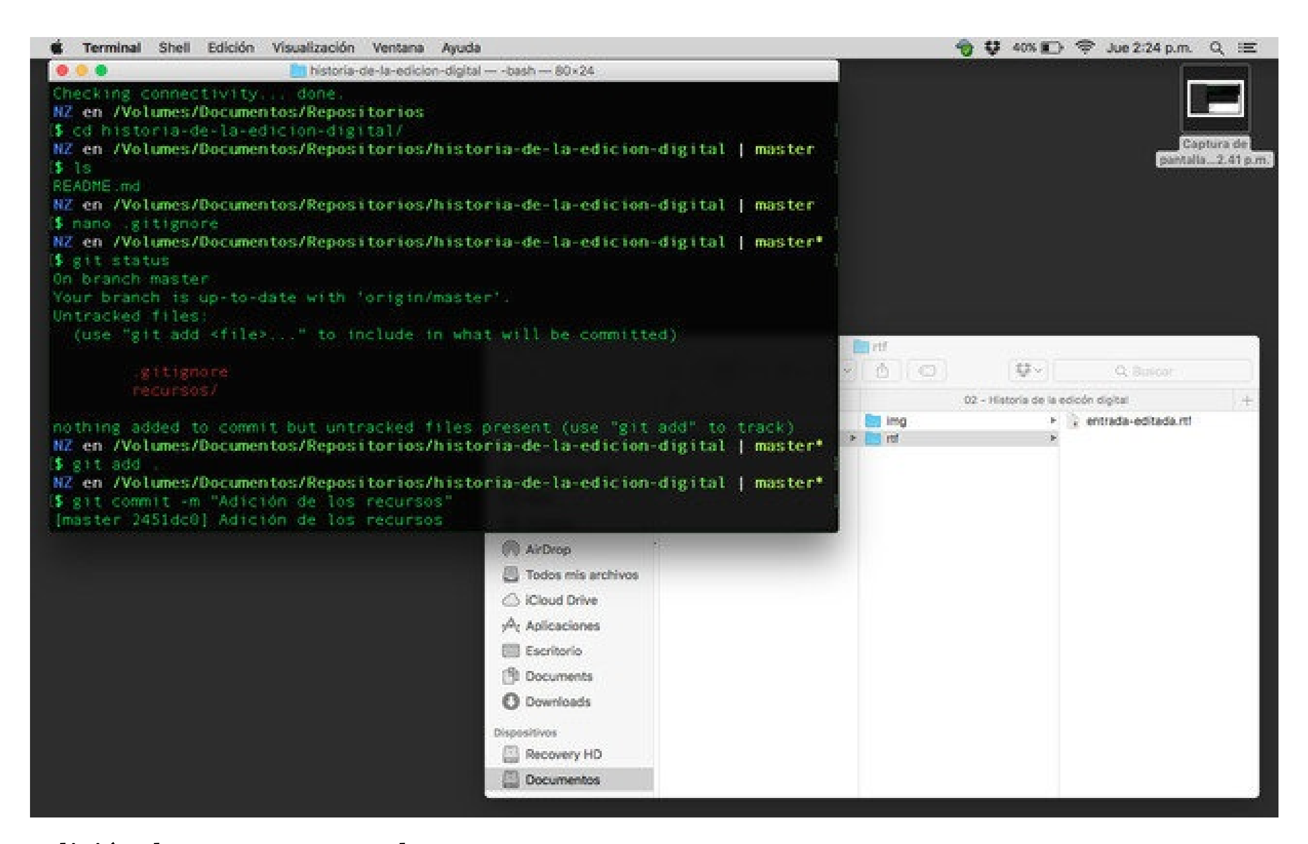
1. tener un gestor de versiones que permita puntos de guardado, evitando el respaldo de archivos cada vez que se termine o se empiece una tarea importante, y,
2. habilitar la conexión a un servidor para salvaguardar el proyecto si la computadora deja de funcionar.

Para este caso utilizamos GitHub como servidor de nuestro repositorio, ya que es una plataforma estable y pública. De esta manera, no tenemos que preocuparnos por el mantenimiento y, en este caso, para que además cualquiera pueda descargarlo.

Por ello, se genera el repositorio en GitHub y a continuación se clona (léase «descarga») a la computadora.

Por supuesto, es posible crear repositorios privados para que solo nosotros o un equipo de trabajo tenga acceso, pero no es lo que se pretende en esta publicación.

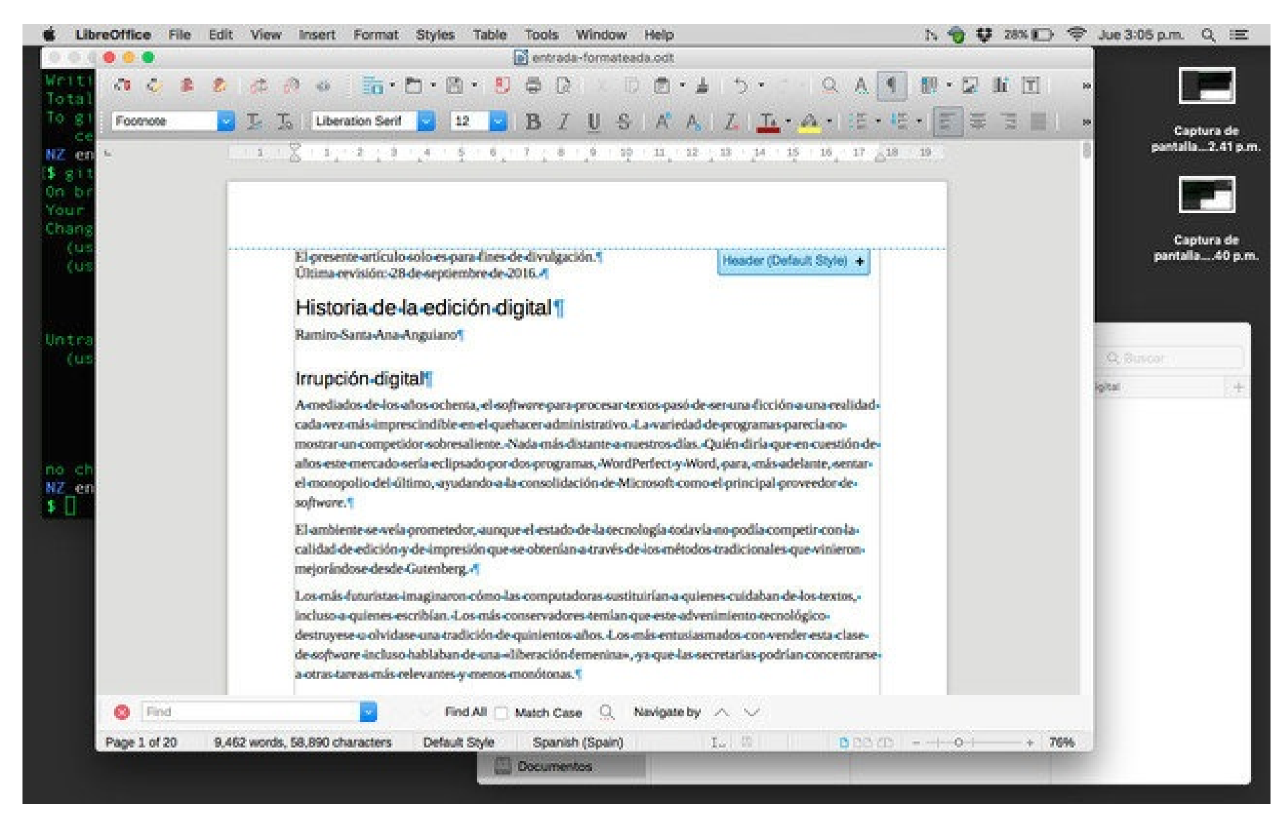
2. Adición de recursos, 14:24 h



Adición de recursos, 14:24 h.

A continuación se añaden al proyecto las imágenes y el texto editado por Mariana. Y, antes de seguir con el siguiente paso, preferí ir a comer porque lo había olvidado, je.

3. Revisión del archivo original, 15:05 h

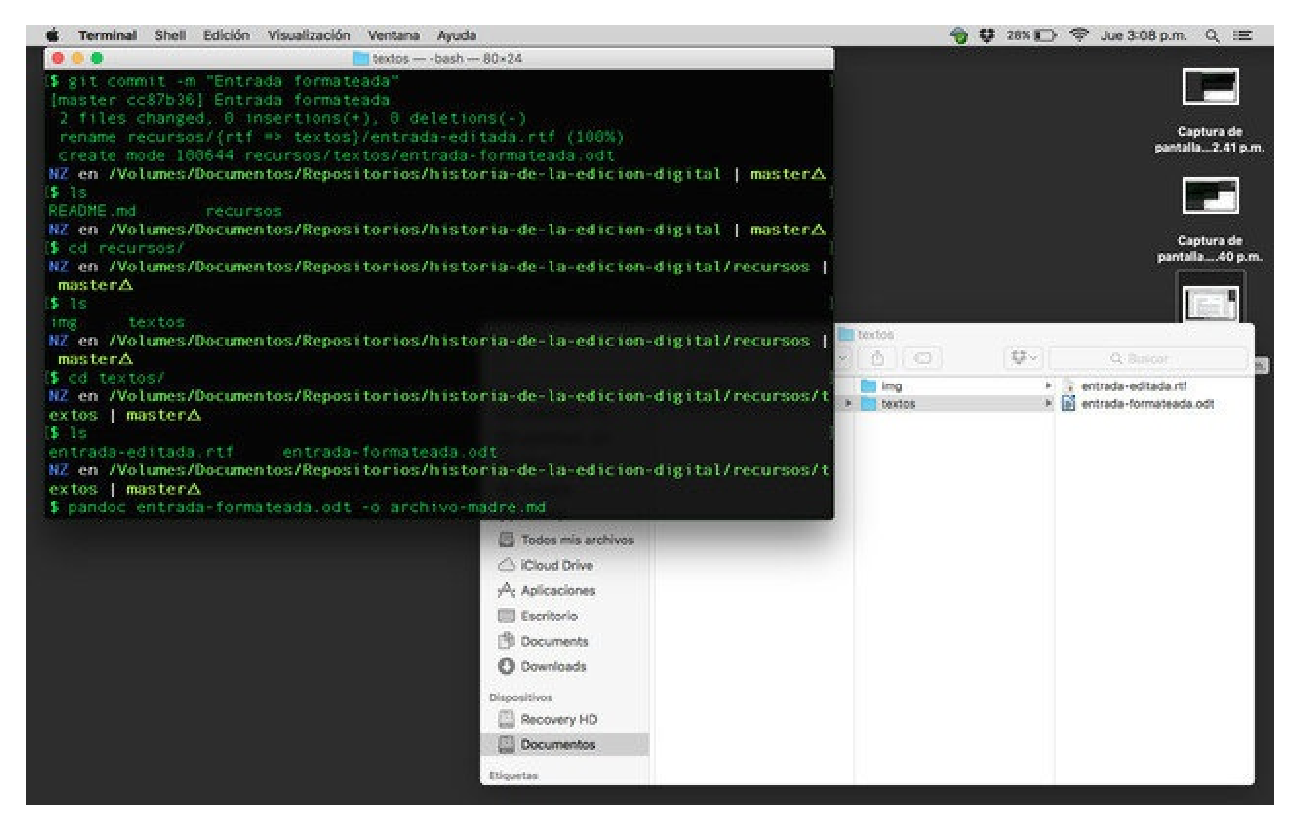


Revisión del archivo original, 15:05 h.

Con Mariana estuvimos editando el texto entre el formato RTF y ODT, ya que no cuento con Microsoft Office ni tenía el deseo de trabajar con los formatos DOC o DOCX.

Entre el RTF y ODT luego suceden cosas «extrañas» en el formato, por lo que se ha de examinar el archivo para constatar que todo está en orden.

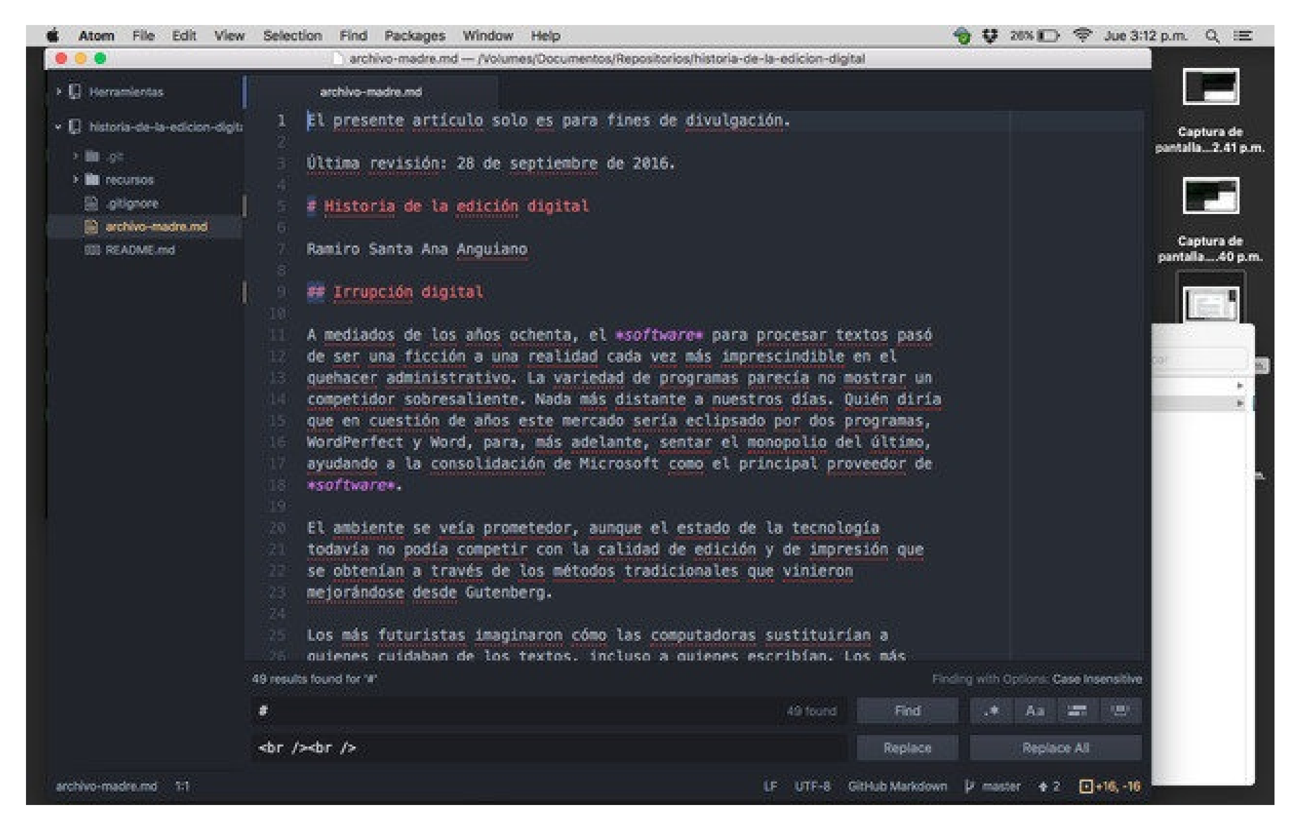
4. Conversión del archivo original, 15:08 h



Conversión del archivo original, 15:08 h.

Mediante Pandoc se convierte el texto original a Markdown. Este proceso es uno de los más breves e importantes ya que así obtenemos el formato más apto para nuestro archivo madre.

5. Revisión del archivo madre, 15:12 h



Revisión del archivo madre, 15:12 h.

Markdown es un lenguaje de marcado ligero, que permite dar estructura al texto (encabezados, bloques de cita o itálicas, por ejemplo) con una sintaxis fácil de aprender y de leer.

La consecuencia de esta versatilidad es, sin duda, una simplificación en la estructura. Markdown es un lenguaje con el que podemos determinar una gran variedad de elementos, pero sus posibilidades se quedan cortas si lo comparamos con HTML o TeX.

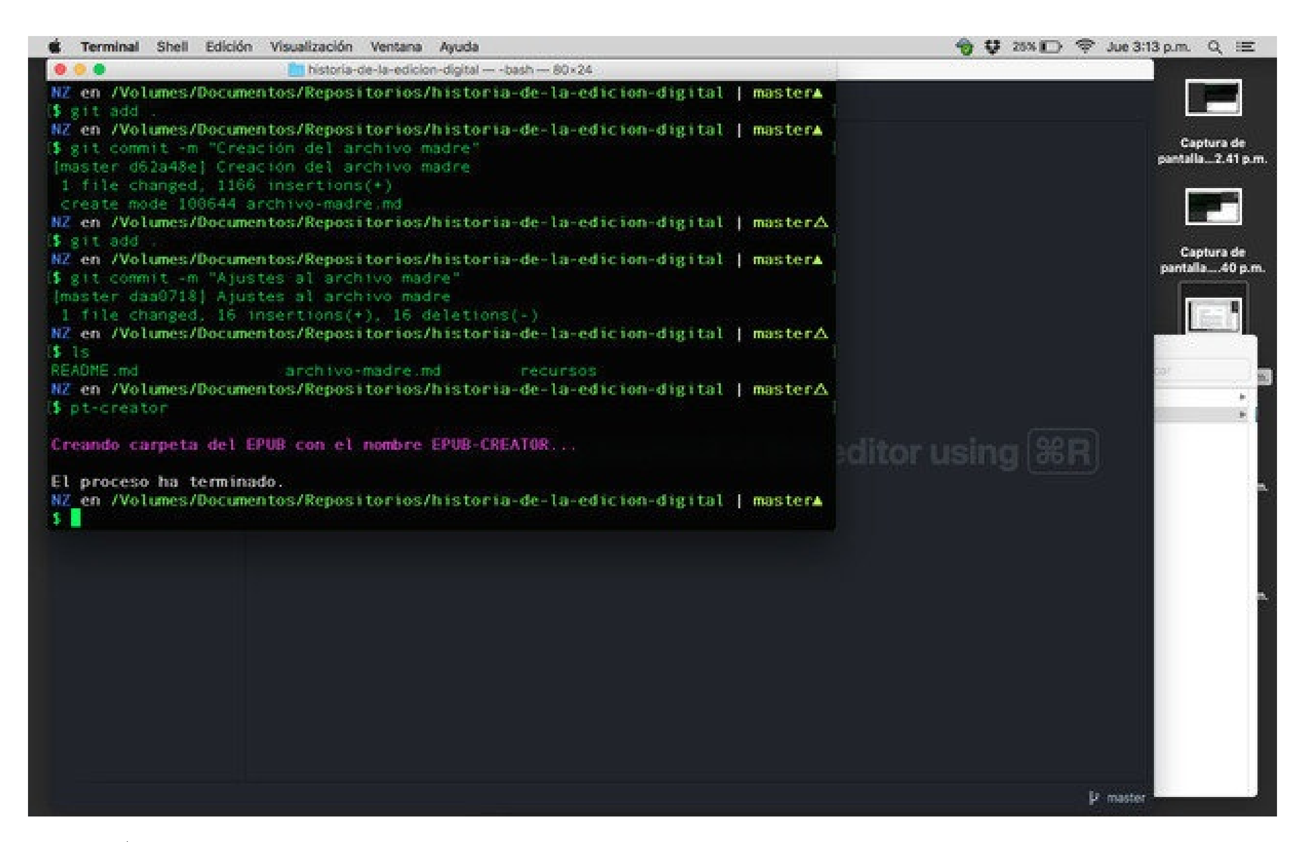
Sin embargo, **es en su simplicidad por lo que Markdown es el lenguaje más óptimo para el archivo «madre»** de cada uno de los formatos de nuestra publicación.

La idea detrás de esto es ir de lo simple a lo complejo, del Markdown a otros formatos que, por su naturaleza, requieren ajustes particulares.

En este paso solo se da un vistazo al archivo madre y es más

adelante donde agregaremos otros elementos.

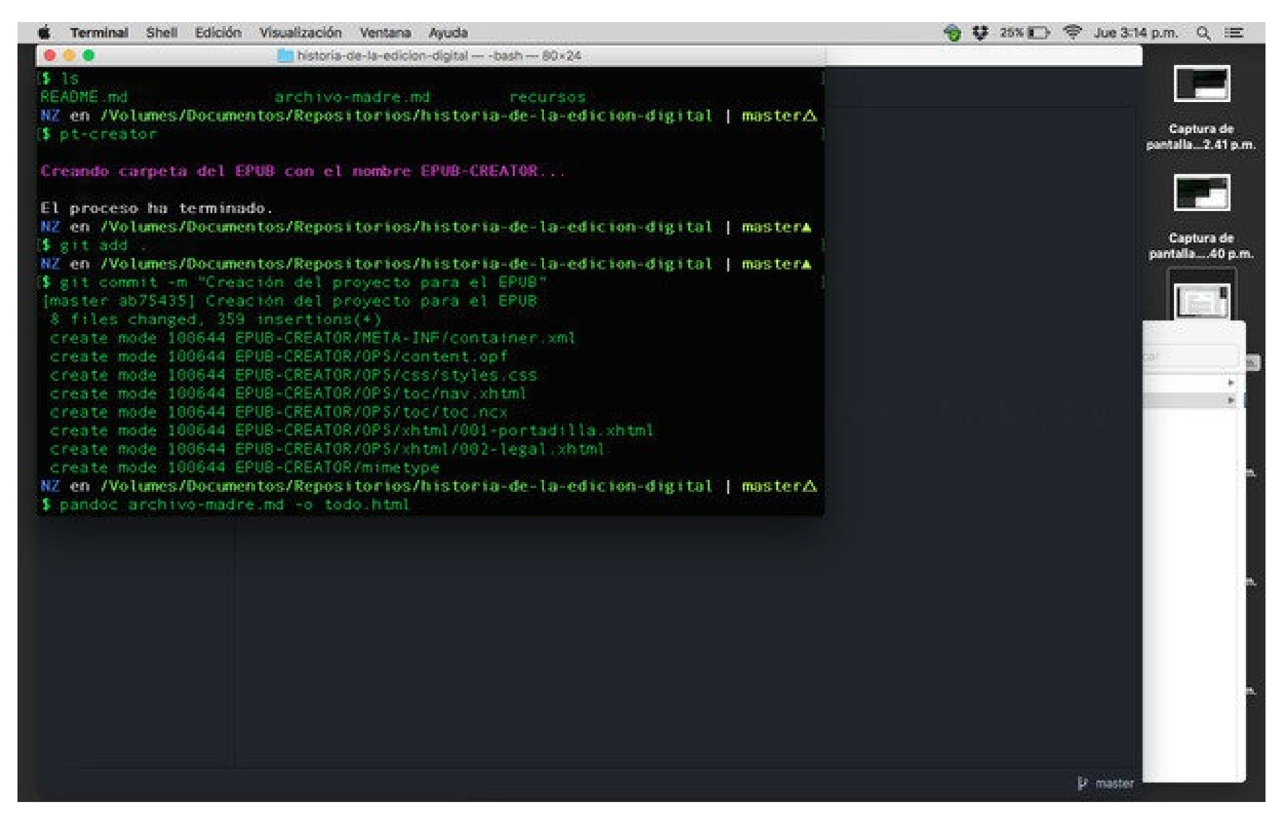
6. Creación del proyecto para EPUB, 15:12 h



Creación del proyecto para EPUB, 15:12 h.

Una de las primeras herramientas de Perro Triste es `pt-creator`, la cual nos crea un proyecto base para un EPUB. Con esto evitamos empezar el proyecto desde cero o a partir de la copia de otro anterior, ya que nos implementa una estructura de archivos y carpetas convencional a cualquier EPUB.

7. Conversión del archivo madre a HTML, 15:14 h

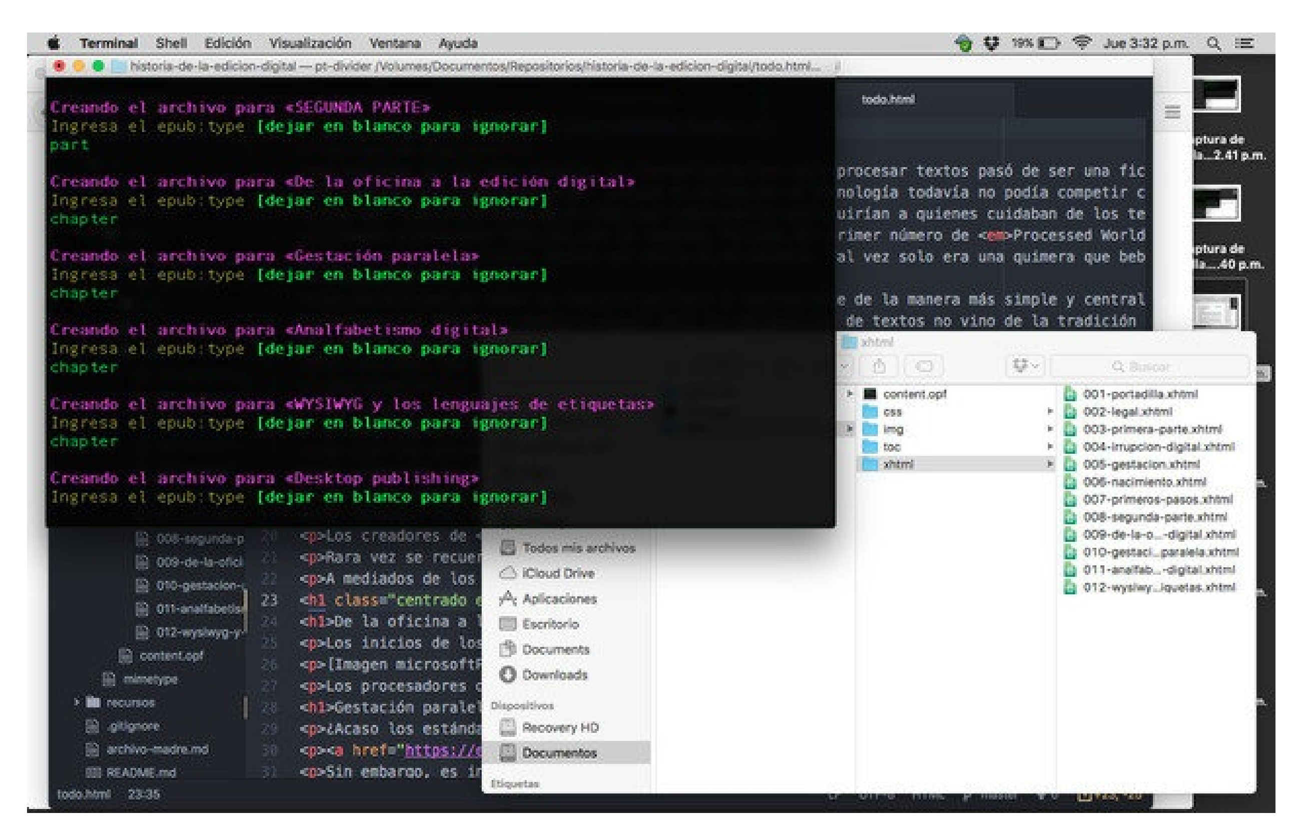


Conversión del archivo madre a HTML, 15:14 h.

Para el EPUB requerimos al menos un archivo HTML, el cual tiene su origen en el archivo madre que convertimos en HTML a través de Pandoc.

En este punto es cuando también se añaden elementos adicionales al HTML, como son las clases de párrafos, los identificadores o las imágenes.

8. División del archivo HTML, 15:32 h



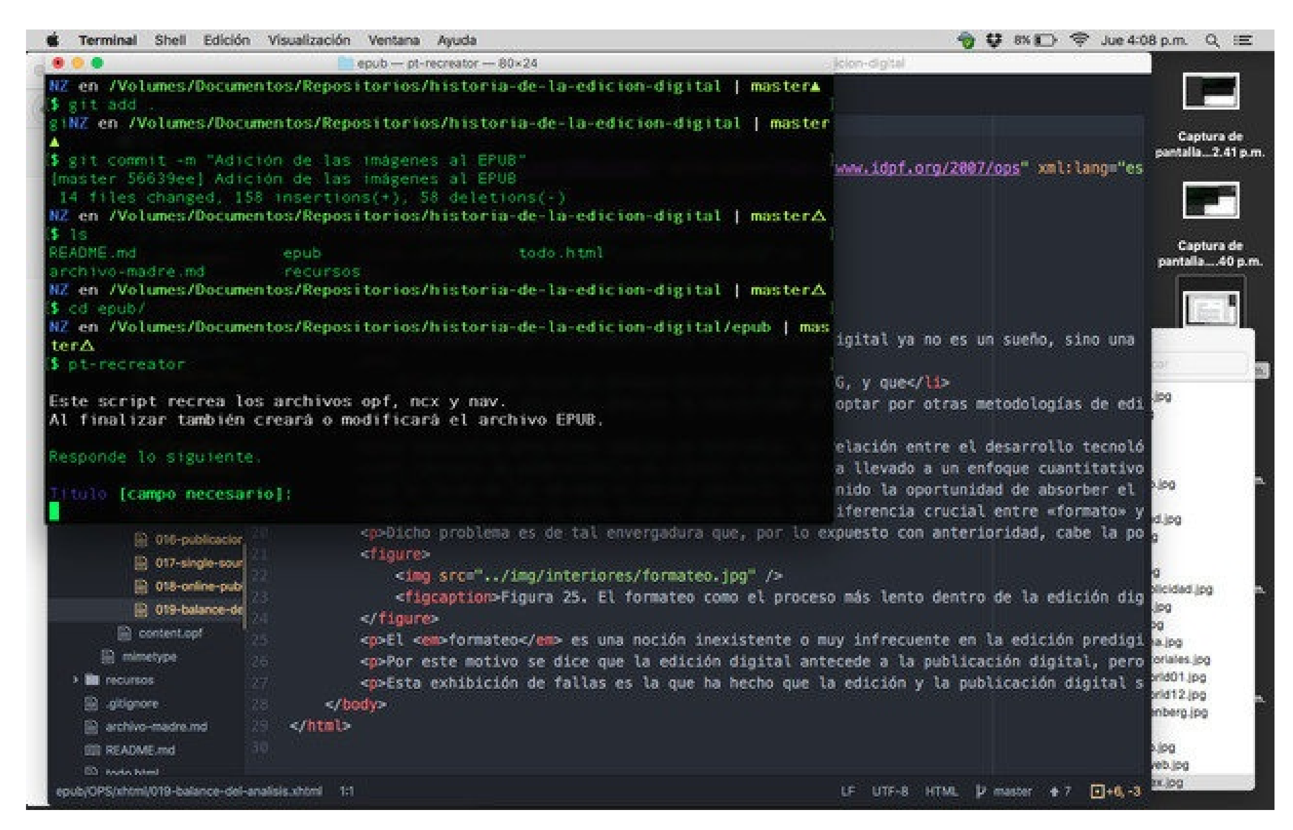
División del archivo HTML, 15:32 h.

Por lo general, un EPUB tiene un documento XHTML por cada sección de la obra, así que ahora dividimos el archivo HTML por medio de p`t-divider`.

Este *script* divide el documento de entrada cada vez que encuentra un encabezado de primera jerarquía (`h1`), permitiéndonos especificar el tipo de sección que se trata y generando automáticamente un XHTML.

Ups, alguien llama a la puerta. Hora de suspender un poco…

9. Creación del EPUB, 16:08 h



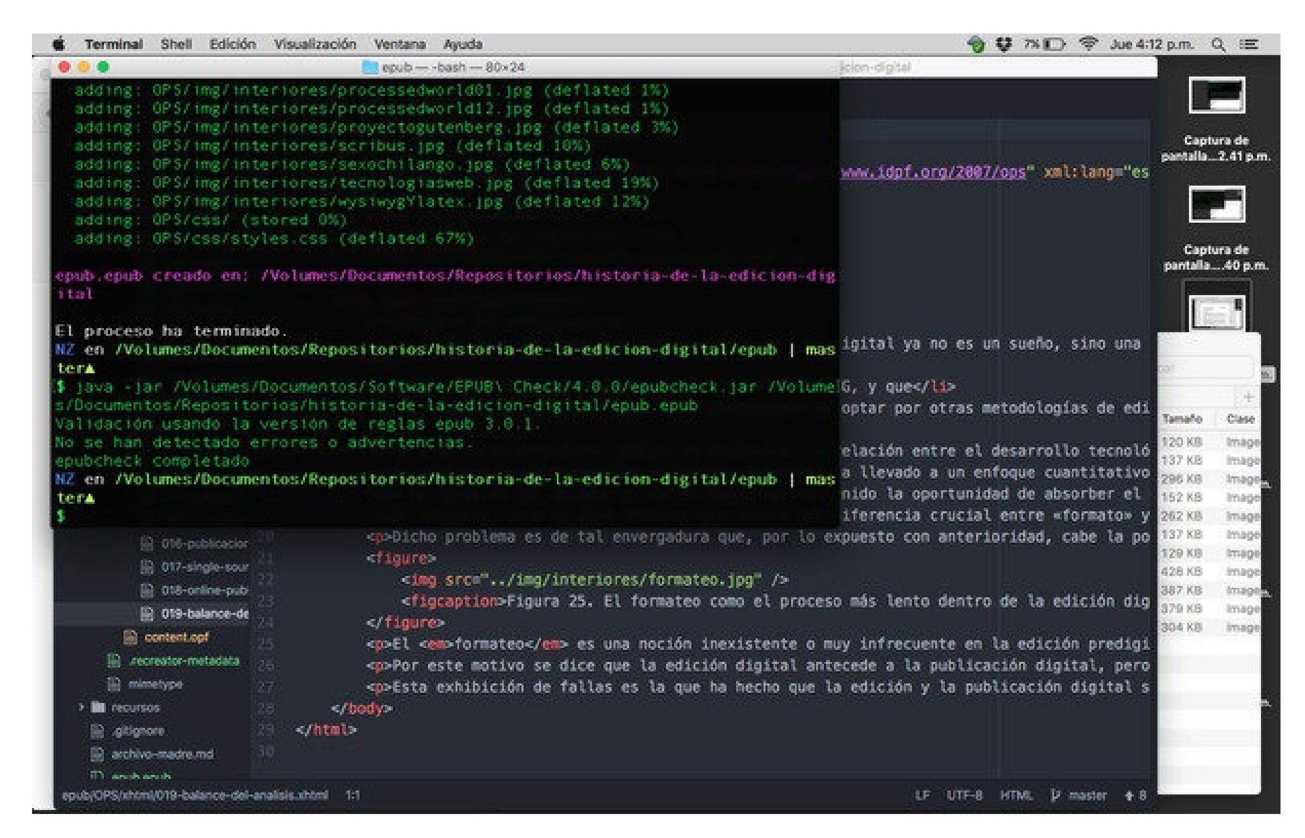
Creación del EPUB, 16:08 h.

Ahora ya contamos con todo lo necesario para nuestro EPUB, por lo que procedemos a crearlo con `pt-recreator`.

Este es uno de los *scripts* más importantes ya que **evita que nos preocupemos de la correcta introducción de metadatos o de la creación de todos aquellos archivos propios del EPUB**, como las tablas de contenidos.

Así pues, poco a poco nos acercamos al ideal donde quien edita o quien diseña puede centrarse exclusivamente en su trabajo.

10. Validación del EPUB, 16:12 h



Validación del EPUB, 16:12 h.

Crear el EPUB es una cuestión, pero otra es desarrollar correctamente un EPUB. Para saber si nuestro *ebook* tiene coherencia interna, utilizamos Epubcheck, el validador oficial.

Como `pt-recreator` automatiza la creación de la «médula» del EPUB, los posibles errores o advertencias que nos indique Epubcheck estarán relacionados con nuestra labor de marcado o de inclusión de recursos. (En ocasiones bien específicas algún error es ocasionado por una herramienta de Perro Triste que se van depurando conforme se van identificando).

Si el EPUB tiene errores, solo basta con corregirlos y regenerar de nuevo el EPUB con `pt-recreator`, un proceso tan sencillo que incluso este *script* guarda los metadatos ya introducidos.

Para evitar la gran mayoría de los errores, recomiendo dos cosas:

1. **usar formatos XHTML en lugar de HTML**, ya que así no se permiten ambigüedades en las etiquetas, y
2. **ver siempre los archivos en Firefox antes de crear el EPUB**, debido a que este explorador libre no te muestra el archivo si tiene errores, caso contrario a Safari o Chrome.

11. ¡EPUB listo! 16:14 h

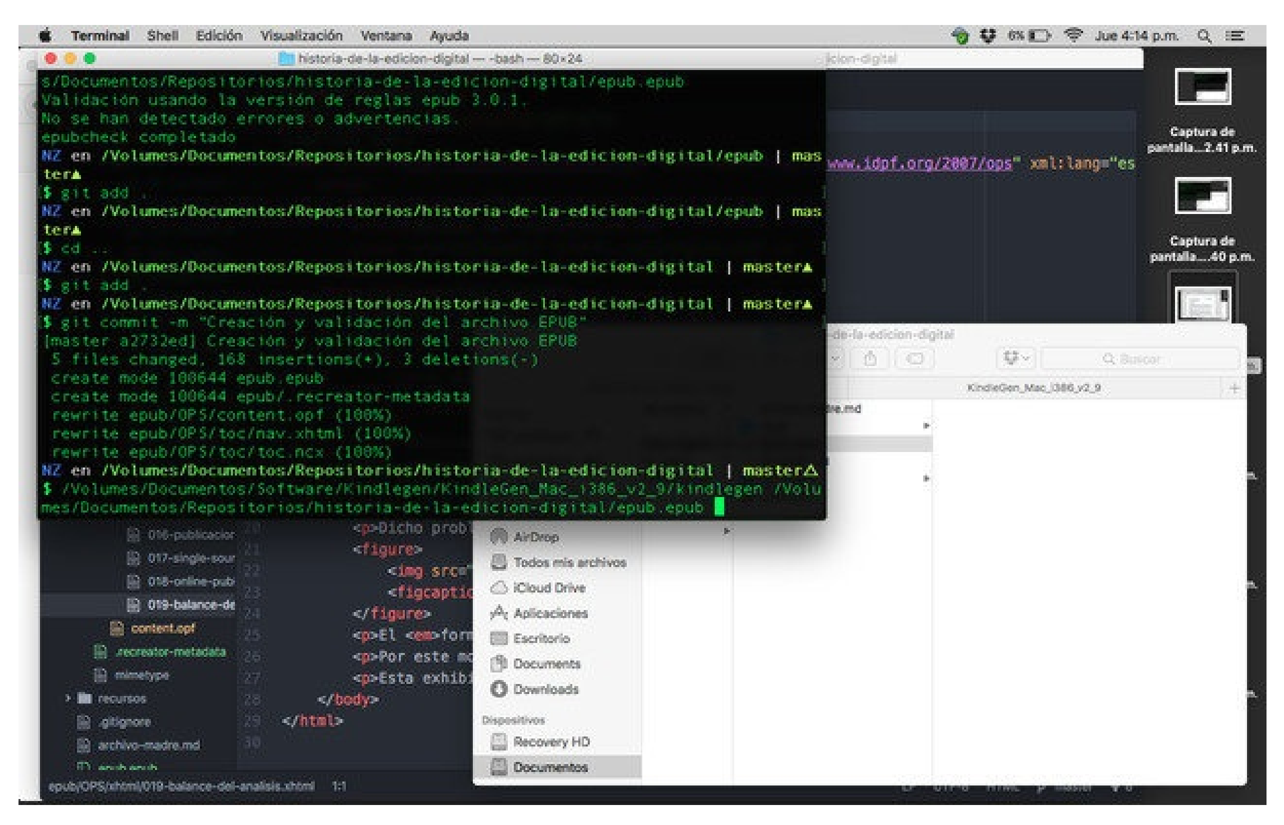


¡EPUB listo! 16:14 h.

En menos de dos horas terminamos el EPUB, sin errores y con un gran control sobre la estructura y el diseño y eso que una hora se nos fue entre la comida y la llamada a la puerta.

¿Acaso el proceso podría simplificarse y estar más enfocado a las necesidades editoriales? ¡Sí, es en lo que estamos trabajando!

12. ¡Conversión del EPUB *a* MOBI*! 16:14 h*

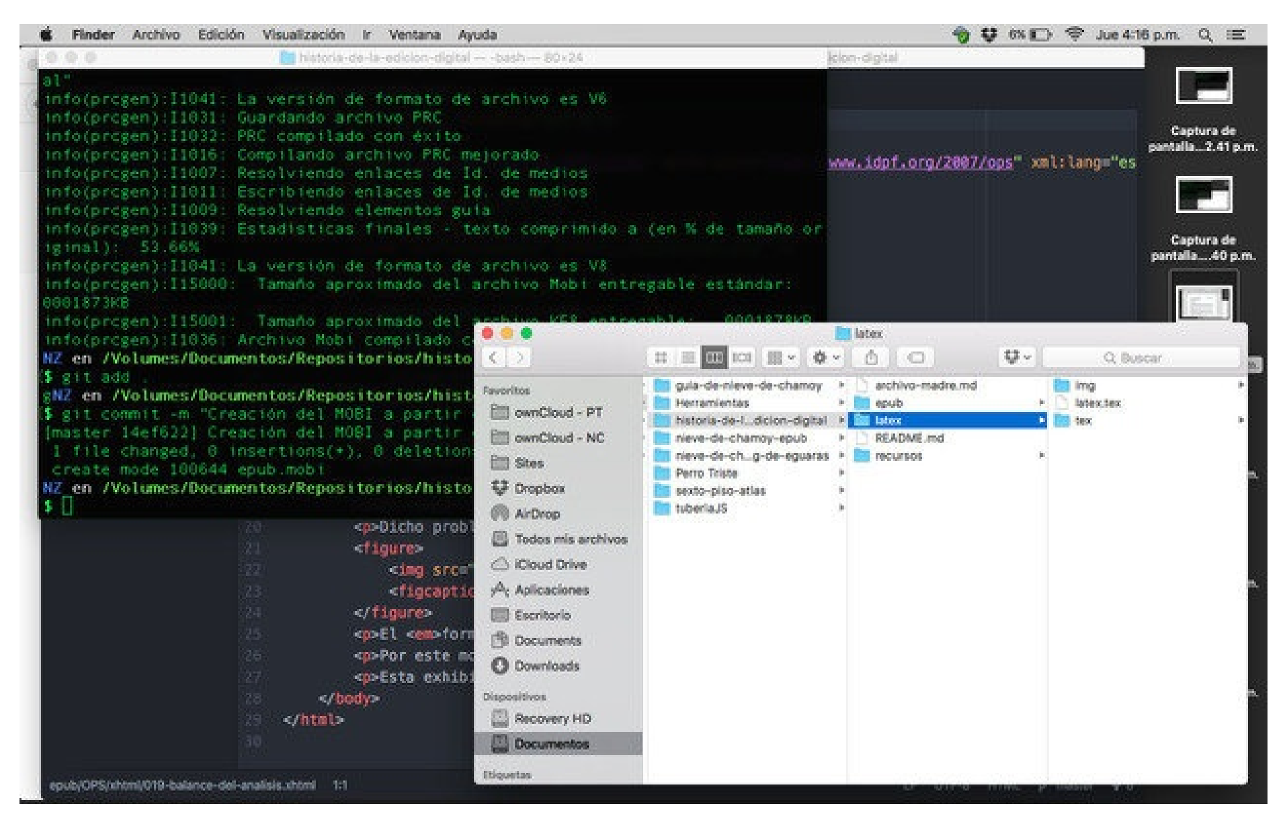


¡Conversión del EPUB a MOBI! 16:14 h.

Para el archivo legible para Kindle no tenemos que hacer gran cosa. Amazon nos pone KindleGen a nuestra disposición, una herramienta que nos permite convertir el EPUB a su formato privativo.

¡En menos de un minuto ya tenemos listo el *ebook* para Kindle!

13. Creación del proyecto para LaTeX, 16:16 h

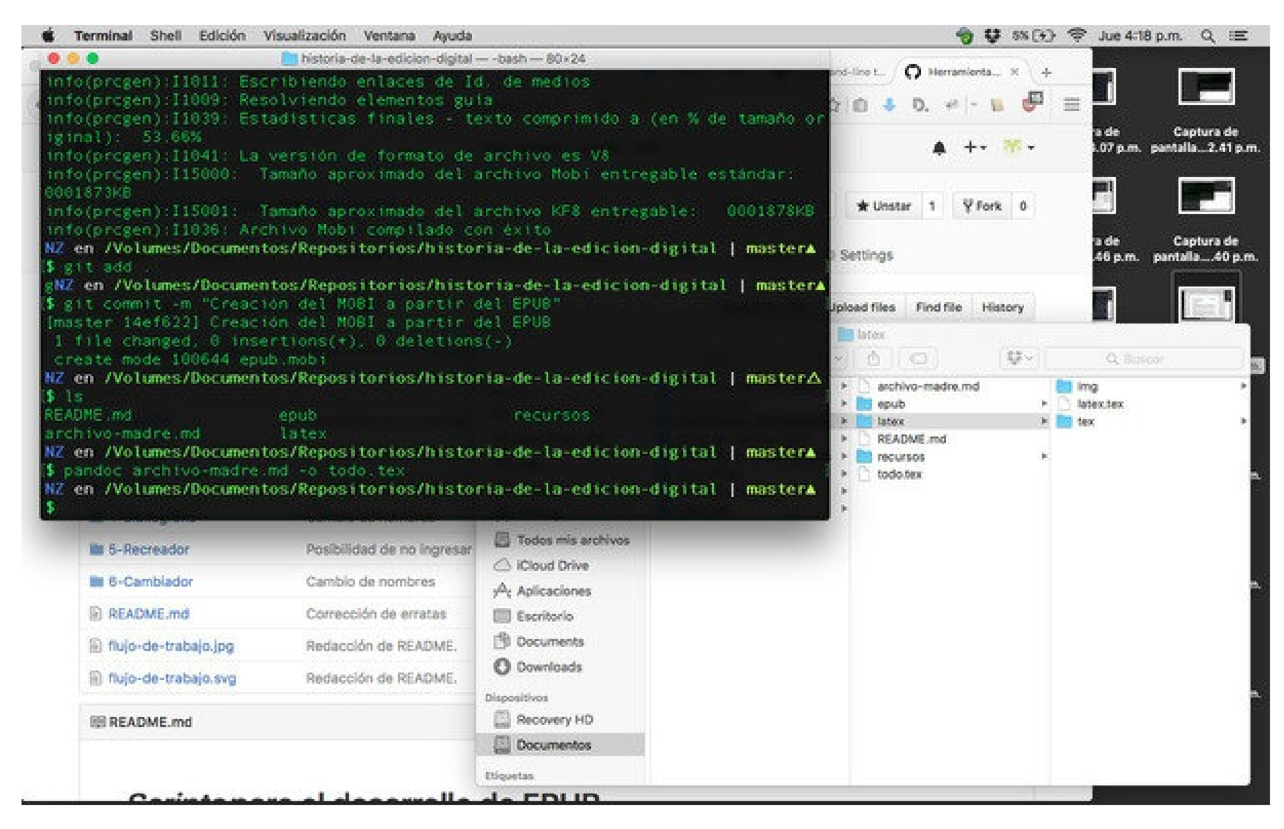


Creación del proyecto para LaTeX, 16:16 h.

Ahora es momento de concentrarnos en la **creación del PDF**. Para esto no usaremos el *software de facto* en el mundo editorial (InDesign), sino una herramienta más antigua y potente: TeX.

Para facilitarnos la creación del EPUB, emplearemos un conjunto de macros conocido como LaTeX. Por el momento, no existe un *script* de Perro Triste para crear un proyecto base de LaTeX. Por tanto, se copia y pega manualmente una plantilla que se encuentra en el repositorio de estas herramientas.

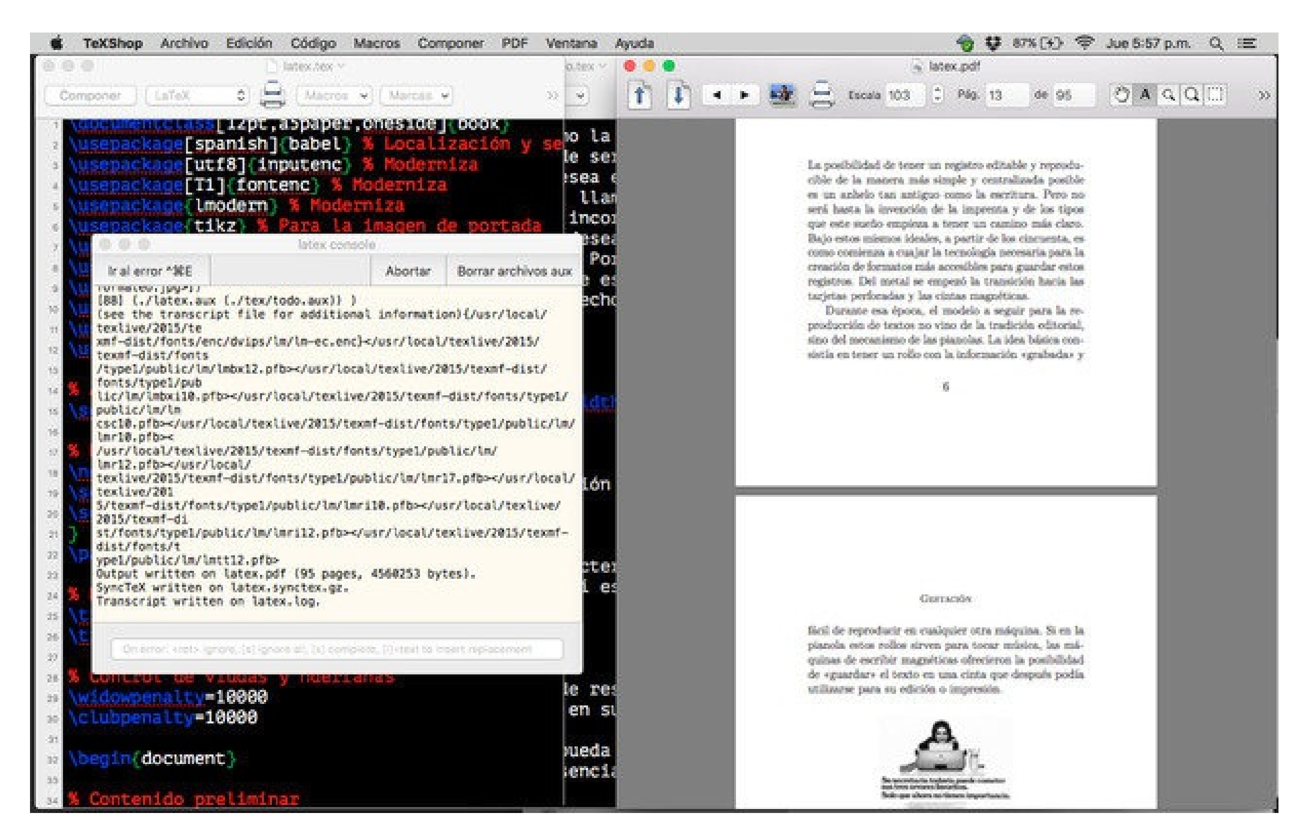
14. Conversión del archivo madre a TeX, 16:18 h



Conversión del archivo madre a TeX, 16:18 h.

De nueva cuenta, convertimos el archivo madre a través de Pandoc, pero ahora para tener un formato TeX. A partir de aquí se empieza la labor de cuidado editorial característico de un impreso, por lo que nos tomamos un tiempo para corregir cajas, viudas, huérfanas, etcétera.

15. ¡PDF listo! 17:57 h

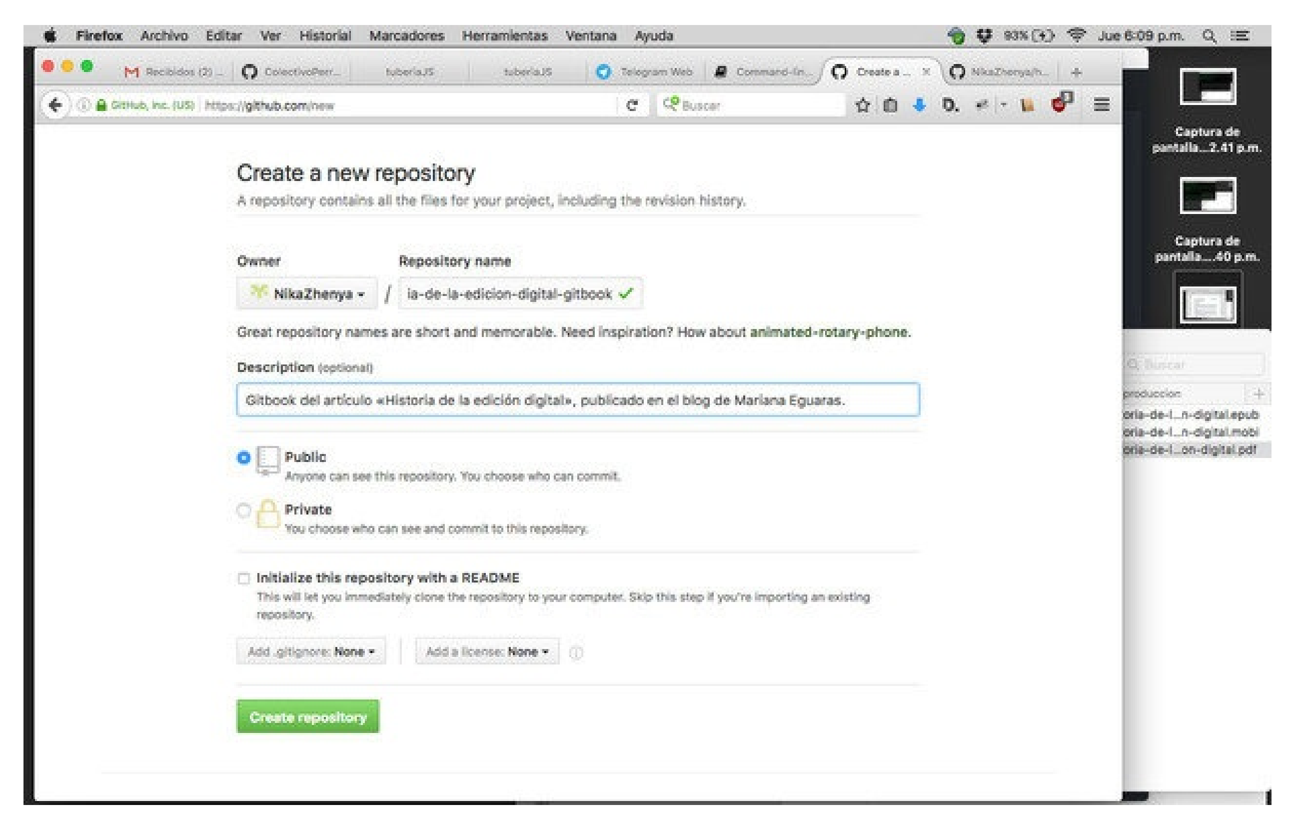


¡PDF listo! 17:57 h.

Después del meticuloso cuidado en LaTeX (ni tanto, porque Mariana encontró varios detalles al PDF), **¡obtuvimos un PDF en poco más de hora y media!**

El PDF puede modificarse para que sea una salida apta para impresión, pero en este caso solo optamos por la posibilidad de una lectura digital.

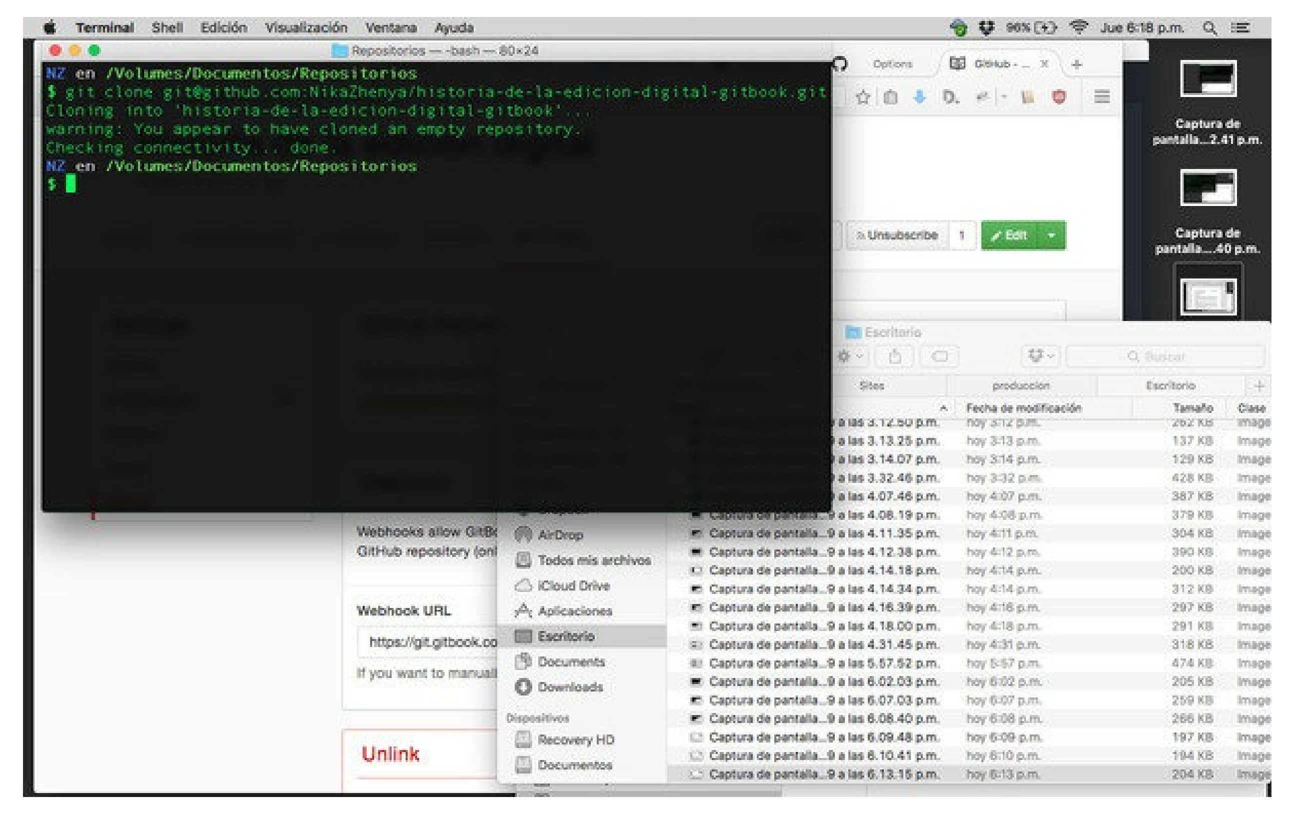
16. Creación del proyecto para GitBook, 18:09 h



Creación del proyecto para GitBook, 18:09 h.

Por último, también quisimos tener una opción completamente *online*. Para esto optamos por GitBook, un proyecto creado por GitHub, inicialmente para la creación de documentación de *software*.

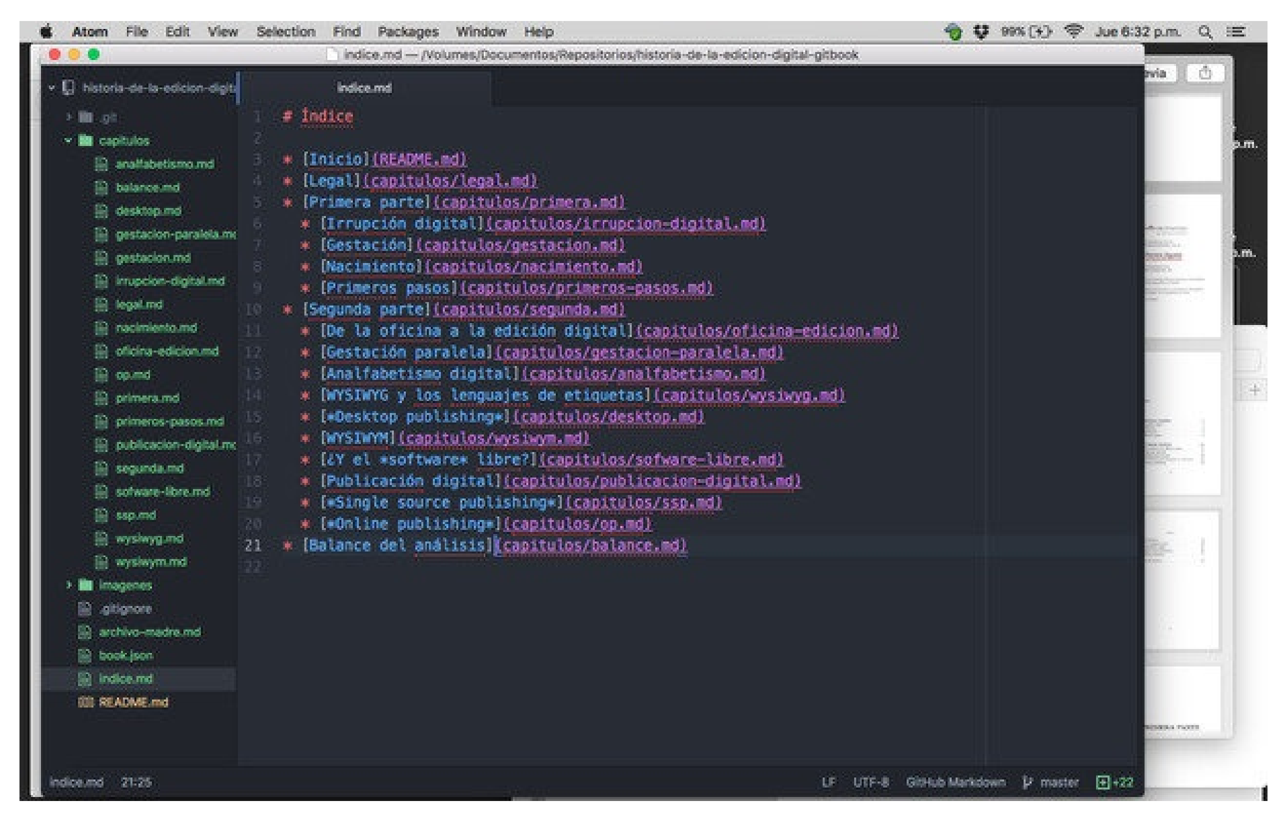
17. Clonación del repositorio de GitBook, 18:18 h



Clonación del repositorio de GitBook, 18:18 h.

Una vez que configuramos el libro en GitBook, procedemos a descargar el repositorio. En este punto es donde la publicación comprende dos repositorios, uno para el formato de GitBook y otro para el resto de los formatos. Esto se debe a que GitBook actualiza automáticamente la publicación cuando se mandan cambios al servidor.

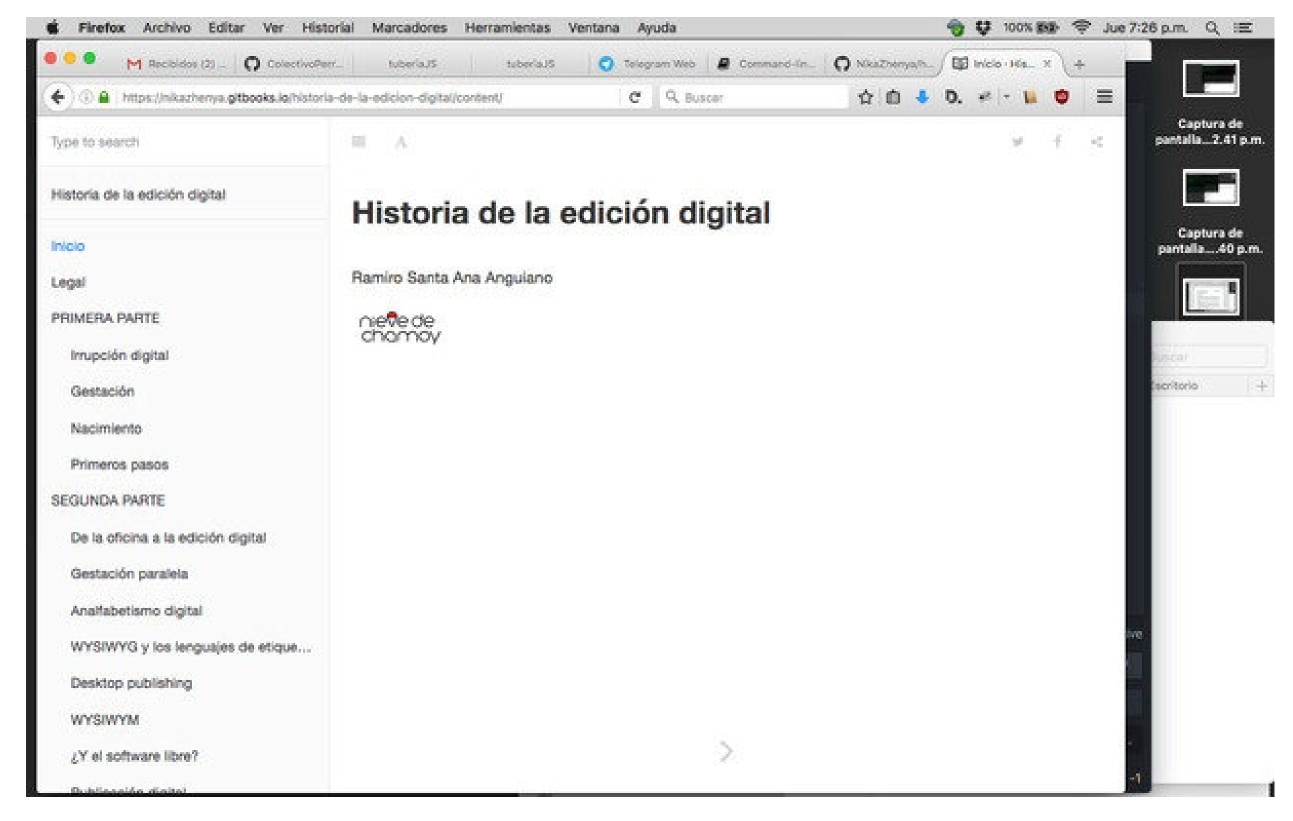
18. División del archivo madre, 18:32 h



División del archivo madre, 18:32 h.

El formato que utiliza GitBook es Markdown, por lo que solo es necesario crear una serie de archivos en este formato para copiar y pegar cada una de las secciones. Posteriormente, en otro Markdown se crea la tabla de contenidos y ya tenemos todo para publicar este formato.

19. ¡GitBook listo! 19:26 h



¡GitBook listo! 19:26 h.

Después de poco más de una hora, el artículo ya está disponible en línea cuando se mandan los cambios al servidor.

En un lapso de cinco horas desarrollamos cuatro formatos del *ebook*, que cualquier persona puede descargar o leer en EPUB, PDF, MOBI o en línea. Mientras tanto, ¿ya hay hambre de nuevo, no?

Single source y *online publishing*

A partir de un archivo madre se obtuvieron tres formatos de una misma publicación: EPUB, PDF y GitBook. Y a partir del EPUB se creó el formato MOBI. Esto es un ejemplo del *single source publishing*, donde a partir de un tronco en común (en este caso el Markdown) se crearon distintas «ramas» para cada uno de los formatos.

De esta manera, todos tienen un elemento en común al mismo

tiempo contemplan sus particularidades. Y además hacen que estos cambios específicos no afecten al resto de los formatos.

Una vez terminado el desarrollo de los diversos formatos, Mariana cotejó cada uno de ellos para depurar errores. Este proceso de repechaje también exige una atención especial (está en la lista de deseos de Perro Triste). Sin embargo, por el momento nos acotamos a modificar y recrear cada uno de los archivos. Proceso que no consume más tiempo que el primer desarrollo.

En el caso de existir algún problema, el *online publishing* permite recuperar información a partir de puntos de control e incluso tener el texto disponible en línea con una simple actualización.

Por ejemplo, a partir de ahora desde los distintos enlaces pueden descargarse o leerse el artículo con la seguridad de que siempre se accederá a la versión más actualizada, ¡porque no tenemos que copiar y pegar archivos en otro lugar!

Por las imágenes mostradas, más de uno pensará que se trata de procesos complejos para la mayoría de las personas que publican un libro…

Lo único que puedo mencionar es que ese es el motivo por el que **en Perro Triste estamos desarrollando estas herramientas**, así como en Nieve de Chamoy es donde se prueban y afinan en el día a día del quehacer editorial.

Al final, la gran pregunta es: **¿qué otras posibilidades existen para facilitar y automatizar la creación de diversos formatos con tiempos semejantes a un día de jornada laboral?**

Existe la convicción de que quien edita o quien diseña no debe dejar de lado el cuidado que exigen sus profesiones. Pero, para que esto suceda, debe haber un empoderamiento tecnológico que convierta las actuales exigencias del mundo editorial en

posibilidades de mejora en lugar de dificultades.

Sobre esta entrada

- Título original: «Tras bambalinas de “Historia de la edición digital”».
- Título en el *blog* de Mariana: «Cómo generar 4 formatos de un contenido en un día».
- Fecha de publicación: 8 de febrero del 2017.
- Redacción original.
- Revisión de Mariana.

Derecho de autor cero: entre el derecho de autor y el dominio público

La siguiente propuesta surge ante la falta de certeza jurídica *post mortem* de las obras abiertas, desde el fallecimiento del autor hasta su traslado al dominio público. Se pretende explicar una alternativa legal para mediar entre el carácter conservador de las actuales legislaciones del derecho de autor y el deseo de muchos autores de que sus obras estén disponibles para todo el público.

Esta propuesta no busca rivalizar con las licencias de uso u otras alternativas del quehacer cultural en pos de la apertura de la información. En su lugar, busca complementarlas al centrarse en el problema de la protección que reciben las obras una vez que los autores han fallecido.

Problema

Antes de la muerte del autor

En la actualidad, los autores que no están de acuerdo con la legislación vigente de derecho de autor cuentan con la alternativa de emplear licencias de uso. Estas licencias surgieron en el ámbito del *software* libre y de código abierto, buscando promover la libertad de uso del material creado por uno o más autores. Por este motivo la noción de *software* libre ha sido el punto de partida para otros movimientos como la «cultura libre» o el «acceso abierto».

Dentro del campo que abarca el derecho de autor —obras literarias o artísticas; programas de radio o televisión, y *software*, entre otros— existe una gran cantidad de propuestas de licencias de uso, tales como las licencias Creative Commons (CC), la Licencia de Producción de Pares o la Licencia Editorial Abierta y Libre (LEAL).

Entre los distintos tipos de licencias el concepto de «libertad» es variable, en algunas ocasiones ambiguo e incluso conflictivo. No todas las licencias de uso cumplen las mismas necesidades y en ocasiones *restringen* ciertos usos, resultando contraproducentes para la libertad en la utilización de una obra. Cabe mencionar que una característica básica de las licencias de uso es que permiten el empleo «gratuito» del contenido. En este sentido es preciso indicar que la gratuidad no necesariamente significa libertad —el *copyleft* sirve para poder distinguir entre una y otra—.

Sin embargo, la mayoría de los autores optan por las licencias de uso a partir de otras cuestiones:

1. comodidad, ya que el trámite es tan sencillo como llenar un formulario en internet —y a veces ni eso—, en contraste con los procesos administrativos necesarios para el registro tradicional de una obra;
2. economía, porque el registro tradicional por lo general requiere cuotas de recuperación, y
3. difusión, al no haber necesidad de pedir permiso o pagar por el uso de una obra, el usuario será más proclive a difundir el trabajo del autor.

La relación establecida por las licencias de uso entre el autor y el usuario cambia la dinámica cultural. Ante este reto, diversas legislaciones han sido modificadas para que semejantes licencias

tengan cabida dentro el ámbito jurídico. Como lo ha indicado cc de manera clara, no existe una rivalidad entre las licencias y el derecho de autor, sino una apertura de «todos los derechos reservados» a «algunos derechos reservados».

Esto ha ayudado a evidenciar un fenómeno recurrente en la producción cultural: el autor tiene que luchar constantemente para hacer valer su voluntad —ya sea de apertura o de cierre—. Aunque el derecho de autor fue constituido precisamente para salvaguardar los intereses del autor, este se ve obligado en el día a día a tolerar ciertas prácticas para tener acceso a una infraestructura cultural en pos de la «libertad» creativa, de la calidad de su obra o de una difusión más amplia.

Muchos de los autores que desean la apertura de su obra tienen que enfrentarse al hecho de que la mayoría de las convocatorias o becas para la producción y difusión de su trabajo implican cierta pérdida de libertad por parte de los usuarios. Es por ello que en ocasiones se recurre a alternativas para sustentar la producción abierta de contenidos, tales como trabajar una jornada adicional, los donativos o el micromecenazgo.

No obstante, esta tensión supone que el autor está vivo. En el plano legislativo esto se conoce como actos jurídicos *inter vivos*. Es decir, el autor tiene al menos la posibilidad de establecer relaciones que busquen concretar sus intereses. Pero ¿qué pasa cuando el autor fallece? En el caso habitual los herederos continúan con el control de los derechos patrimoniales de la obra, «control» que en muchas ocasiones también fue voluntad del autor. Sin embargo, ¿qué sucede cuando la voluntad del autor es la apertura de su obra?

Después de la muerte del autor

Las licencias de uso no sustituyen el derecho de autor, sino que facilitan su reproducción. Por ello, el derecho de autor continúa vigente a pesar del empleo de licencias y, también, después de la muerte del autor.

Por *convención* el fallecimiento del autor no implica el traslado inmediato de su obra al dominio público. Bajo el argumento de salvaguardar los intereses de los herederos o de los productores-editores, hay un lapso de al menos cincuenta años —establecido por el Convenio de Berna— para que la obra al fin sea de acceso y uso libre. No obstante, la prolongación del derecho de autor suele ser mayor; por ejemplo, en la Unión Europea es de setenta años y en México de cien años.

Debido a que muchos de los autores mueren intestados, o sus personas cercanas desconocen los actos jurídicos que fueron celebrados en vida, se crea una ambigüedad jurídica que dura de cincuenta a cien años. Este lapso de tiempo impide la oportuna reproducción de la obra, generando lo que se conoce como «obras huérfanas» —creaciones cuyos titulares de derechos no son conocidos o localizables—.

Solo los grandes consorcios tienen la capacidad legal de producir-editar obras con vacíos jurídicos. Para la mayoría de los productores-editores independientes, la apuesta por la reproducción de una obra huérfana implica el riesgo de enfrentarse a situaciones legales que sobrepasan sus alcances económicos. En estos casos, las obras huérfanas que no representan altas ganancias —como pueden ser las obras de culto o para un público muy específico— quedan sin ser publicadas durante varios años.

Pero esta no es la única dificultad: en muchos casos la voluntad del autor no es respetada por sus herederos. No son pocas las

situaciones en que estos —sea una institución o algún familiar— son más reservados que su testador al momento de establecer acuerdos para publicar. Este carácter conservador llega al grado de no permitir la reedición de la obra o mantenerla inédita debido a que los herederos intentan obtener el mayor beneficio económico, aumentando los costos de producción —conflicto alimentado por las abusivas prácticas editoriales— y en detrimento de la conservación de la herencia cultural.

Para el caso de los autores que en vida expresaron el deseo de apertura de su obra, este problema se agudiza. También por *convención* el dominio público no es una adopción voluntaria, sino restringida a las pautas de tiempo mencionadas con anterioridad. Por ello, en muchas legislaciones no existe un medio jurídico para poder trasladar de inmediato una obra al dominio público, aunque la voluntad del autor hubiera sido la apertura de su obra.

La implementación del dominio público voluntario es un debate abierto que seguramente permanecerá así durante más tiempo, ya que en varias legislaciones implicaría una reforma profunda a las leyes del derecho de autor. Esto no implica que tal posibilidad carezca de importancia, sino que mientras la discusión continúe existe la necesidad de crear alternativas, a corto o mediano plazo, para los autores que tienen la disposición de abrir su obra a todo el público.

Estas alternativas no solo han de enfocarse en la creación de licencias uso, ya que el problema no se limita al anhelo de algunos autores por el dominio público voluntario; tampoco a la búsqueda de mayor flexibilidad en los derechos reservados, dentro de un contexto jurídico que continuamente está fortaleciendo al derecho de autor. La dificultad también reside en la concreción de este deseo tras la muerte del autor.

En muchas legislaciones, las licencias de uso tienen un carácter temporal. El permiso de uso o difusión de la obra no es perpetuo: se encuentra limitado a cierto tiempo que, de no ser explícito, cada legislación contempla por defecto. Esto implica que los herederos de los derechos pueden adquirir de nuevo todos los derechos reservados del testador. Como se mencionó antes, las licencias no sustituyen al derecho de autor; esto se traduce en que *las licencias de uso no son suficientes para mantener la apertura de una obra por tiempo indefinido*.

No se debe pasar por alto que, ante esta situación, también existen alternativas que apuestan por un quehacer cultural más democrático, transparente y comunitario. Sin embargo, mientras el ámbito legislativo no fomente o proteja este tipo de prácticas, cualquier iniciativa de apertura de la información que no incida en la capa legal de la infraestructura cultural carece de garantía para el futuro.

Nuevamente: no por ello carecen de importancia estas iniciativas; por el contrario, son relevantes para mostrar otras posibilidades de quehacer cultural, aunque por ahora sean insuficientes para resolver el problema jurídico de hacer valer la voluntad de apertura expresada por un autor fallecido.

Propuesta

Entidad de gestión

Para salvaguardar el deseo de los autores de mantener abierta su obra una vez que hayan fallecido, es necesaria una red de organizaciones o instituciones que velen por este interés. Estas

instituciones han de recibir obras de manera voluntaria, quedando supeditadas a los autores que coincidan en la importancia de la libertad de las personas usuarias para la difusión de su obra y la conservación de la herencia cultural.

Para el eficaz funcionamiento de estas organizaciones se ha de partir desde una dimensión nacional. El Convenio de Berna solo brinda pautas mínimas para la gestión internacional del derecho de autor, provocando que cada legislación presente diferencias sutiles al respecto. Ante tal situación, es necesario que estas instituciones partan del contexto jurídico de su Estado correspondiente, para que cada una se constituya como una entidad de gestión de derechos de autor —«sociedad de gestión colectiva» para el caso mexicano—.

Esto no implica que las distintas instituciones nacionales no puedan organizarse para fomentar lazos de cooperación. Por el contrario, estas relaciones internacionales son necesarias, ya que las problemáticas desatadas por el derecho de autor —y la propiedad intelectual en general— no se acotan a un marco nacional. Lo importante es que su surgimiento sea a partir de las *reglas de juego* de cada nación.

Para evitar los problemas recurrentes dentro de las políticas culturales, estas entidades han de ser públicas, descentralizadas, autónomas y con un carácter civil. Además, para facilitar la apertura de la obra, el trabajo administrativo del día a día ha de realizarse en la medida de lo posible con formatos abiertos y *software* libre o de código abierto —medidas características de las «instituciones abiertas»—. Asimismo, requerirá el apoyo de otras organizaciones para el asesoramiento en la conservación de obras culturales libres y estandarizadas, como puede ser el seguimiento a las pautas indicadas en la Definición de Obras Culturales Libres

(Definition of Free Cultural Works).

Esta red de organizaciones ha de tener como objetivos en común:

1. La difusión abierta y gratuita de las obras bajo su resguardo, al igual que su conservación.
2. La implementación de mecanismos digitales sencillos que permitan al autor ceder los derechos de su obra.
3. La creación de una estructura tan flexible y democrática como sea posible.
4. La emisión pública y constante de toda la información concerniente a la administración.
5. El fomento a la investigación interdisciplinaria en materia de derecho de autor y propiedad intelectual.
6. El desarrollo y la cooperación tecnológica necesarios para la consecución de los objetivos previos.

De esta manera, el autor que desee que su obra permanezca abierta por tiempo indefinido puede optar por ceder los derechos patrimoniales a estas instituciones sin fines de lucro. Ahora bien, estas serían las principales diferencias entre estas organizaciones y las entidades de gestión tradicionales:

- La gestión entraría en vigor una vez fallecido el autor, para mantener la flexibilidad de uso de su obra sin necesidad de pedir autorización.
- El uso de la obra sería gratuito y no necesitaría un permiso por escrito ni un aviso a las instituciones.
- Las organizaciones no emprenderían acciones legales, *excepto* cuando los derechos morales sean violados.

Debido a la simplificación administrativa, la flexibilidad jurídica y el desarrollo tecnológico en pos de la facilidad de uso para autores y usuarios, estas instituciones presentarían diferencias que se sintetizan con la concepción del «derecho de autor *cero*»:

- cero pagos,
- cero autorizaciones con antelación,
- cero comunicaciones previas,
- cero complejidades administrativas,
- cero dificultades de uso y
- cero medidas legales.

En cuanto a la capa jurídica, estas organizaciones habrían de erigirse como entidades de gestión de derecho de autor según las pautas indicadas en cada legislación correspondiente. Al indicar la voluntad del autor de ceder sus derechos patrimoniales a esta red cuando fallezca —lo que se conoce como acto jurídico *mortis causa* —, estas organizaciones tendrían los derechos de las obras en un plano legal. Sin embargo, en la práctica estas instituciones no se desempeñarían de modo punitivo, sino que procurarían la difusión abierta y gratuita de las obras hasta que estas al fin formen parte del dominio público.

En este sentido, la relevancia de estas entidades sería la evasión del vacío jurídico existente en la prolongación del derecho de autor hasta su traslado al dominio público. Del mismo modo, evitaría que la voluntad de apertura del autor sea ignorada por los posibles herederos, como pueden ser otras instituciones habituales dentro de la gestión de derechos e incluso los mismos familiares.

Donativos y regalías

A diferencia de las entidades de gestión tradicionales, las instituciones del derecho de autor cero buscarían ser financiadas mediante donativos. Diversas organizaciones vinculadas al *software* libre o de código abierto, a la cultura libre, al acceso abierto y a la protección de la privacidad de los usuarios han demostrado que es un modelo económico factible.

No es la intención que los colaboradores de tiempo completo vivan con los lujos y comodidades a las que están acostumbradas algunas de las personas pertenecientes a la industria cultural —ejecutivos de alto rango o grandes productores-editores de consorcios internacionales, por ejemplo—. Pero esto tampoco implica que los colaboradores deban vivir en situación precaria.

La donación de trabajo sería primordial para el correcto funcionamiento de estas organizaciones. Sin embargo, en la medida de lo posible, se habrían de buscar modelos que al menos permitan la retribución de una renta básica para los colaboradores de tiempo completo. De esta manera se pretendería mejorar sus condiciones de vida y la concentración de esfuerzos en el mantenimiento de las instituciones.

Los donativos podrían ser otorgados por cualquier organización o persona, con la explícita indicación de que esto no permitirá influir en las decisiones tomadas dentro de las instituciones. El *lobbying* estaría absolutamente prohibido. Una medida para evitar el cabildeo sería mediante la publicación de todas las personas o entidades donadoras y los montos concedidos a cada entidad.

Las personas o instituciones que deseen utilizar alguna de las obras bajo resguardo de las instituciones del derecho de autor cero podrán tomarlas de manera gratuita, como se mencionó con anterioridad, o mediante un donativo voluntario. Estas donaciones deberán estar en una lista independiente para que así el usuario

pueda identificar quiénes han donado para la utilización del material disponible. Incluso habría que examinar la viabilidad de que se incluya información referente al contenido producido a partir de las obras utilizadas.

Ahora bien, muchos de los autores también desearán apoyar económicamente a otros herederos, por lo cual tendrá que existir la infraestructura adecuada para que esto sea posible. En estos casos, el autor que ceda su obra a estas organizaciones deberá poder indicar un porcentaje por concepto de regalías para otras instituciones o personas de su preferencia. Así, del donativo obtenido por el uso de su obra —no de los donativos hechos directamente a estas organizaciones— se daría el porcentaje indicado a los herederos correspondientes. Los porcentajes de regalías no deberían exceder el cincuenta por ciento, para permitir una relación equitativa entre el mantenimiento de estas instituciones y el apoyo al resto de los herederos.

Con estas medidas de financiamiento y administración se buscaría un trato justo entre el autor, los usuarios, las instituciones del derecho de autor cero y otros posibles herederos. Es primordial que el autor no interprete que su obra quedaría «secuestrada» por estas organizaciones, por lo que la modificación del acuerdo celebrado por este y la institución nacional correspondiente ha de ser lo más breve posible.

Transparencia

Debido a la estructura e intenciones de las instituciones del derecho de autor cero, toda la información referente a su carácter administrativo deberá estar disponible en línea para cualquier usuario, sin necesidad de registro. En especial, la manera en que se

utiliza el dinero debería ser información abierta y lo más detallada posible: ingresos, lista de donaciones directas, lista de donaciones por la utilización de obras, lista de colaboradores con renta básica, lista de regalías otorgadas, lista de gastos, etcétera.

Sería ideal también buscar una administración que *libere pronto y continuamente*. Así como en el ámbito del *software* existe este esquema de desarrollo, cabría examinar la posibilidad de contar con organismos administrativos y de transparencia que liberen sus contenidos de esta manera, para evitar un mal aprovechamiento del tiempo laboral. O bien, que al menos la información sobre pagos se publique cada semana, quincena o mes. En este sentido, un gestor de versiones podría ser conveniente para poder analizar el historial de modificaciones de estos datos.

Repositorio público

Para la conservación, difusión y uso de las obras, cada institución del derecho de autor cero deberá tener un repositorio público. La búsqueda y descarga de contenidos se haría a través de un sitio en red conectado con todas las bases de datos.

Con la finalidad de alcanzar una mayor difusión y descentralización, los repositorios deberán tener la posibilidad de ser descargados en su integridad o de crear espejos. También sería interesante analizar la viabilidad de subir los contenidos a redes P2P, Library Genesis, Wikimedia Commons o Internet Archive. Aunque parezca descontrolado e incluso contraproducente —pues existe la posibilidad de que un tercero tergiverse la información o utilice las obras para obtener beneficios económicos—, muchos de los autores que cedan su obra tendrían conocimiento de antemano sobre las posibilidades y los peligros que conlleva la

descentralización de la información; aunado a esto, también se buscarían mecanismos que reduzcan las posibilidades mencionadas. Más que un riesgo, para varios podría representar la oportunidad de llegar a plataformas de las que ni siquiera tenían conocimiento y que quizá sean fundamentales para la difusión de su obra.

Los repositorios públicos son tan esenciales que sin ellos la creación de estas instituciones tendría muy poco o ningún sentido. Uno de los principales problemas de las entidades de gestión tradicionales es la reducida accesibilidad a las obras bajo su resguardo, o al menos a los metadatos de las mismas. Incluso al existir una base de datos para su consulta, muchas de estas entidades solo permiten la descarga de contenidos a través de un muro de pago —a veces tan poco organizado que implica una comunicación de ida y vuelta a través de correo electrónico—.

Por este motivo, la infraestructura y la cooperación técnica son primordiales para el funcionamiento de estas organizaciones. Sin embargo, esto no implica un gran desafío técnico. En la actualidad la creación de repositorios públicos, si bien no es una tarea que se deba subestimar, tampoco es de tal complejidad que requiera de un equipo grande de programadores.

Difusión e investigación

Los repositorios, los espejos y la disposición del contenido en otras plataformas no son suficientes para la difusión de las obras. En varios países «en vías de desarrollo», la interacción dentro de comunidades digitales independientes a las redes sociales en boga representa un uso minoritario.

Al partir cada organización, desde contextos nacionales

específicos, la creación de eventos presenciales podría ser una vía de comunicación directa para alcanzar un mayor espectro en cada comunidad. Más que promover la propia organización, estos eventos deberían plantearse como un apoyo para visibilizar diversos prejuicios en torno al derecho de autor y a la propiedad intelectual, mostrando también otras alternativas o puntos de flexibilidad de los derechos de propiedad intelectual.

La intención de estos eventos sería fomentar la discusión y el debate público de algo que se considera propio de los ámbitos jurídicos o especializados —a saber, la legislación correspondiente a la propiedad intelectual—, para evidenciar que en realidad se trata de un tema que concierne a todas las personas y al futuro de la herencia cultural. Por este motivo, la vinculación con otras organizaciones nacionales con posiciones similares a las instituciones del derecho de autor cero, grandes o pequeñas, formales o informales, es necesaria. Estos nexos también deberían generarse de la manera más equitativa posible. El derecho de autor cero no viene a quitarle espacio o importancia a otros movimientos u organizaciones, sino a suplir una carencia específica, como lo es la falta de certeza jurídica *post mortem* de las obras abiertas, desde el fallecimiento del autor hasta su traslado al dominio público.

Para una difusión crítica en torno a los derechos de propiedad intelectual, es menester un organismo independiente de investigación. Con «independencia» se entiende que es preciso realizar investigaciones interdisciplinarias que no dependan de instituciones como la Organización Mundial de la Propiedad Intelectual, el Banco Mundial, el Foro Monetario Internacional, el Centro Regional para el Fomento del Libro en América Latina y el Caribe o las diversas cámaras nacionales de la industria cultural, por poner algunos ejemplos.

Con el fin de evitar que se parta de previos supuestos —como la importancia de la propiedad intelectual para el desarrollo económico—, es necesario analizar no solo el discurso de las instituciones mencionadas en el párrafo anterior, sino también de las organizaciones con las que tienen afinidades e incluso del mismo discurso del derecho de autor cero, como es el caso la presente propuesta. En este sentido, se puede decir que estas organizaciones requieren, en la medida de sus posibilidades, un departamento de investigación que realice una crítica **r**ecursiva y **f**orzosa a todos los actores involucrados en el debate sobre la propiedad intelectual —en otros términos, una metodología de crítica `-rf`—.

Todos los archivos concernientes a estas investigaciones, y no solo el documento final, deberán estar en un repositorio abierto para que cualquiera pueda cotejarlos y analizarlos. Para quienes solo desean los resultados finales habrán de crearse publicaciones en múltiples formatos abiertos o populares, como MD, HTML, EPUB, PDF o MOBI.

Límites

Acto jurídico mortis causa

El derecho de autor cero no busca «competir» con las licencias de uso. Por ello su ejercicio ha de comenzar una vez fallecido el autor. Cuando esto suceda, en ningún caso las instituciones del derecho de autor cero habrán de revocar licencias o demás elementos que el autor haya definido para la difusión de su obra. Únicamente se concentrarán en conservar y difundir la obra una vez que el autor

no se encuentre con vida.

No obstante, valdría la pena discutir la pertinencia de que el ejercicio de estas organizaciones comience en cuanto el autor decida ceder su obra. Un problema grave puede ser que no existan los medios pertinentes para que estas instituciones tengan conocimiento sobre el fallecimiento de un autor. Por este motivo, tal vez el inicio de funciones debería ejecutarse cuando el autor otorgue los archivos de sus obras y de manera paralela a lo que el autor haga para difundir su trabajo. Esta labor no debería interferir con los deseos del autor en ningún sentido.

En el caso de la herencia de derechos a partir del fallecimiento, se celebrará alguna especie de testamento o algún acto jurídico *mortis causa* similar. Esto implica que cada organización nacional requerirá de colaboradores con licencias de abogacía, pero también implica que se han de buscar los mecanismos jurídicos más simples, económicos y eficientes para asegurar esta transmisión de derechos patrimoniales. Quizá en algunas legislaciones la creación de un testamento no sea necesaria; no obstante, siempre se tendrá que contar con un documento legal *ad hoc* para evitar cualquier revés.

Para el posible caso del ejercicio de funciones de las instituciones del derecho de autor cero durante el tiempo de vida del autor, y para evitar que el traslado de derechos patrimoniales impida la publicación de la obra en medios comerciales que tal vez el autor desee, esta organización podría actuar como un agente por el cual tales medios puedan acceder al contenido —como cualquier otro usuario—. En este sentido, la relación no sería entre el autor y el medio comercial, sino mediada por el repositorio de la institución.

El único caso en el que este esquema podría resultar conflictivo

sería si el medio comercial deseara la reserva exclusiva de los derechos. No obstante, es responsabilidad del autor tener conocimiento de que la apertura de su obra no es compatible con la reserva de todos los derechos. En caso de surgir tal dificultad, sería posible plantear que el autor retire la obra de la organización, tal vez con una pequeña penalización debido a los gastos jurídicos requeridos para la celebración de acuerdos legales, aunada a la generación de una lista pública de obras que fueron retiradas por sus autores.

Alternativa, mas no solución definitiva

El derecho de autor cero no pretende imponer su agenda a otras alternativas abiertas llevadas a cabo por algunas organizaciones o personas. Por este motivo es necesario enfatizar que *el derecho de autor cero no es una solución definitiva*, sino una alternativa que se inserta en la capa legal para mediar entre el carácter conservador de las actuales legislaciones del derecho de autor y el deseo de muchos autores de que sus obras estén disponibles para todo el público.

Las instituciones del derecho de autor cero caducarían el día en que la legislación correspondiente tome la decisión de crear mecanismos jurídicos oportunos para la conservación abierta de las obras, sin la necesidad de esperar un lapso de tiempo tras la muerte del autor. Es decir, estas organizaciones cesarían sus actividades cuando las leyes del Estado al que pertenezcan permitan el dominio público voluntario.

Mientras tanto, el derecho de autor cero habrá de ser no solo una idea o un concepto, sino una red de instituciones que puedan manejar la voluntad *post mortem* del autor de mantener su obra

abierta con todos los requerimientos de la ley. Se hace énfasis en la institucionalización del derecho de autor cero por el simple hecho de que es la única medida legal y organizada que permite cumplir con sus objetivos.

Por ello, cuando se habla de instituciones, el paradigma no son las instituciones gubernamentales, de iniciativa privada o partidistas, sino instituciones como Free Software Foundation, Electronic Frontier Foundation, Internet Archive, Document Foundation, Creative Commons, Budapest Open Access Initiative, Wikipedia, Project Gutenberg, SciELO, e incluso el movimiento zapatista o el Congreso Nacional Indígena. En este sentido, el término «institución» se usa en su acepción etimológica de «facilidad», «construcción» y «educación».

Si el derecho de autor cero o el dominio público voluntario son posibles sin la constitución de organizaciones, qué mejor para los autores que desean la certeza jurídica de que sus obras permanecerán abiertas una vez que hayan fallecido: el principal propósito de esta propuesta.

Sobre esta entrada

- Título original: «Derecho de autor cero».
- Título en el *blog* de Mariana: «Derecho de autor cero: entre el derecho de autor y el dominio público».
- Fecha de publicación: 24 de mayo del 2017.
- Repositorio.
- Revisión de Mariana.

Notas

Edición cíclica y edición ramificada: un vuelo seguro y constante

[1] Se ha insistido en el juego «revolución-apertura» ya que la «revolución» en la era digital no siempre se traduce el cambio del *status quo* sino el advenimiento de otros actores más «cerrados». *El código es ley*, por lo que no solo es necesario pensar la era digital como «revolución» sino asimismo como «cierre» o «apertura». Tal vez una historia de la edición desde un planteamiento de «revoluciones» que implican «cierres» y «aperturas» sería interesante…

Cómo las herramientas analíticas pueden mejorar el quehacer editorial

[1] Un falso positivo se da cuando una palabra es marcada como errónea, aunque en realidad sí es una palabra existente, pero no contenida en el diccionario.

Índice analítico